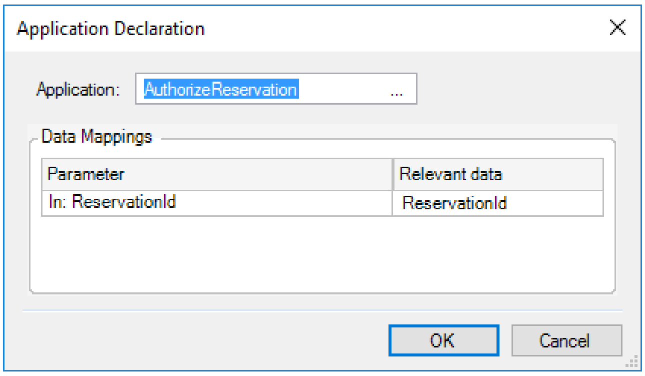
Programming reminders / End of the ticket reservation process

We go back to the diagram to focus on the reservation authorization task, which is carried out by the customer care manager.

In the diagram, we associate the task **Authorize reservation – Customer Care Manager** with the **AuthorizeReservation** web panel, mapping the relevant data item &ReservationId.


Application Declaration

Application: AuthorizeReservation

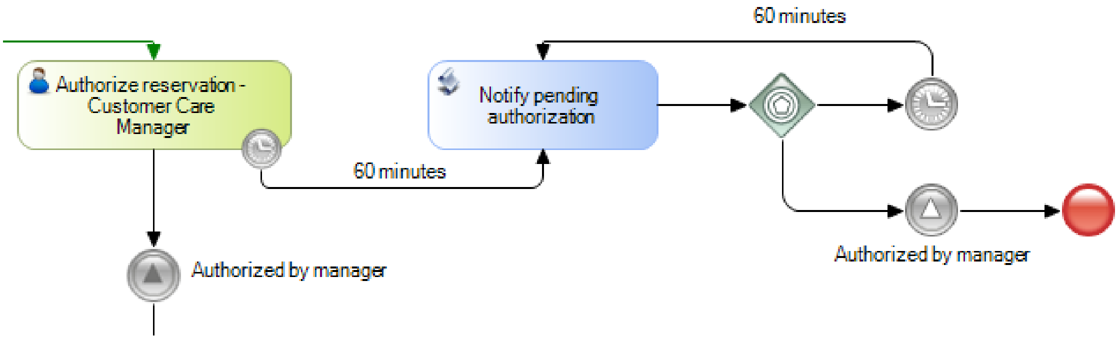
Data Mappings

Parameter	Relevant data
In: ReservationId	ReservationId

The web panel will show the reservation details and will offer 2 buttons to authorize or reject it, similarly to what we saw for evaluating the customer.

We click on OK and create a relevant data item **&AuthorizeReservation** to save the decision made by the customer care manager.

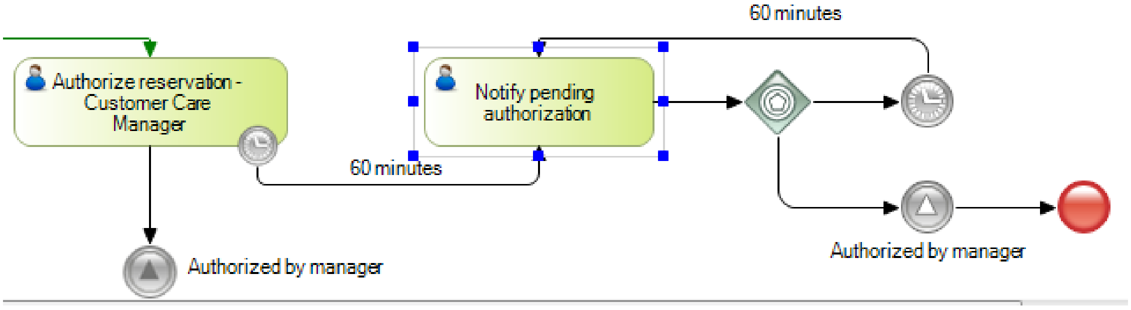
Now, let’s remember the automatic notifications that we had created.


60 minutes
Authorize reservation - Customer Care Manager
Notify pending authorization
60 minutes
Authorized by manager
Authorized by manager

The task **Notify pending authorization** is a batch task that will run a procedure in which a notification is sent to the customer care manager, for example, via email.

In order to make it easier to check this pattern, we will replace the batch task with an interactive task associated with a web panel that simply shows the notification on screen.

To do so, we open the task's properties, change its **Type** property to User and associate the task with the web panel “**NotifyPendingAuthorization**”.


60 minutes
Authorize reservation - Customer Care Manager
Notify pending authorization
60 minutes
Authorized by manager
Authorized by manager

We've changed the waiting times to 2 minutes in order to show this behavior in this video, so we edit the rule property of both timers and enter value 2.

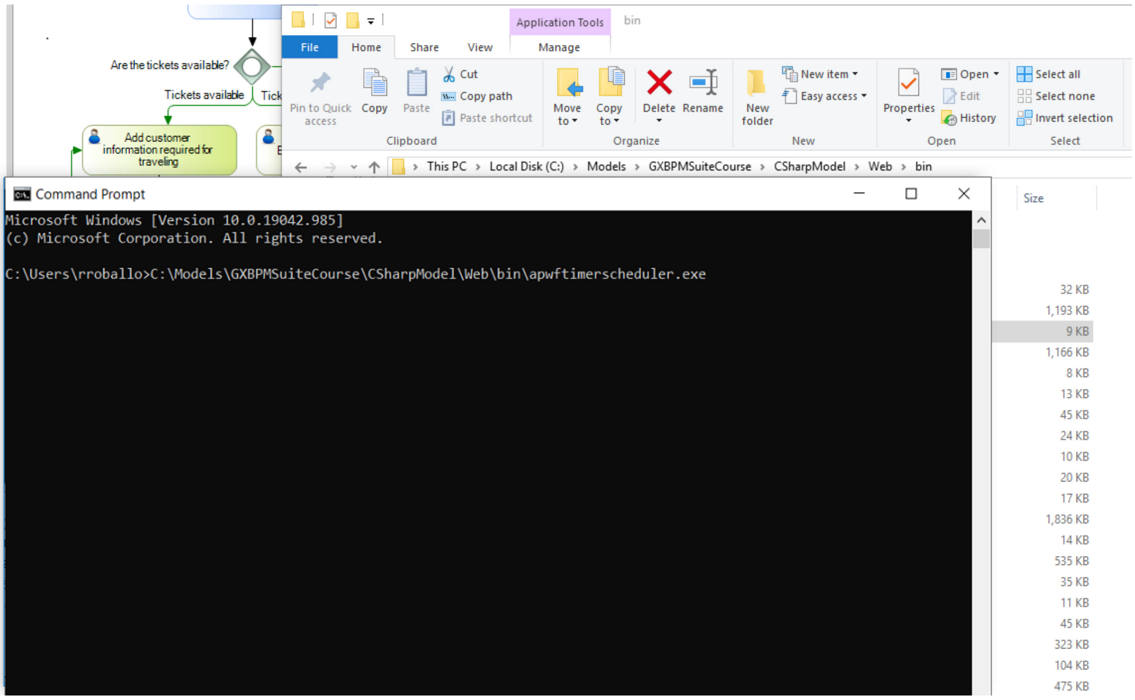
To display the actions caused by timer type events in our inbox, we need to run a program that triggers the events of timer type; if we don't run it, these events will not be triggered.

To execute it, you must do the following:

https://wiki.genexus.com/commwiki/servlet/wiki?43454,Timer%20Control

This utility is run from a command line, so we click on Tools, Explore Target Environment Directory and click on the bin folder.

We open a command window and from the Windows explorer window we drag the executable file **apwftimerdeadlinescheduler.exe**. We enter the value 60 and press Enter. In our case, deadlines will be triggered every 60 minutes.



```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rroballo>C:\Models\GXBPMSuiteCourse\CSharpModel\Web\bin\apwftimerscheduler.exe
```

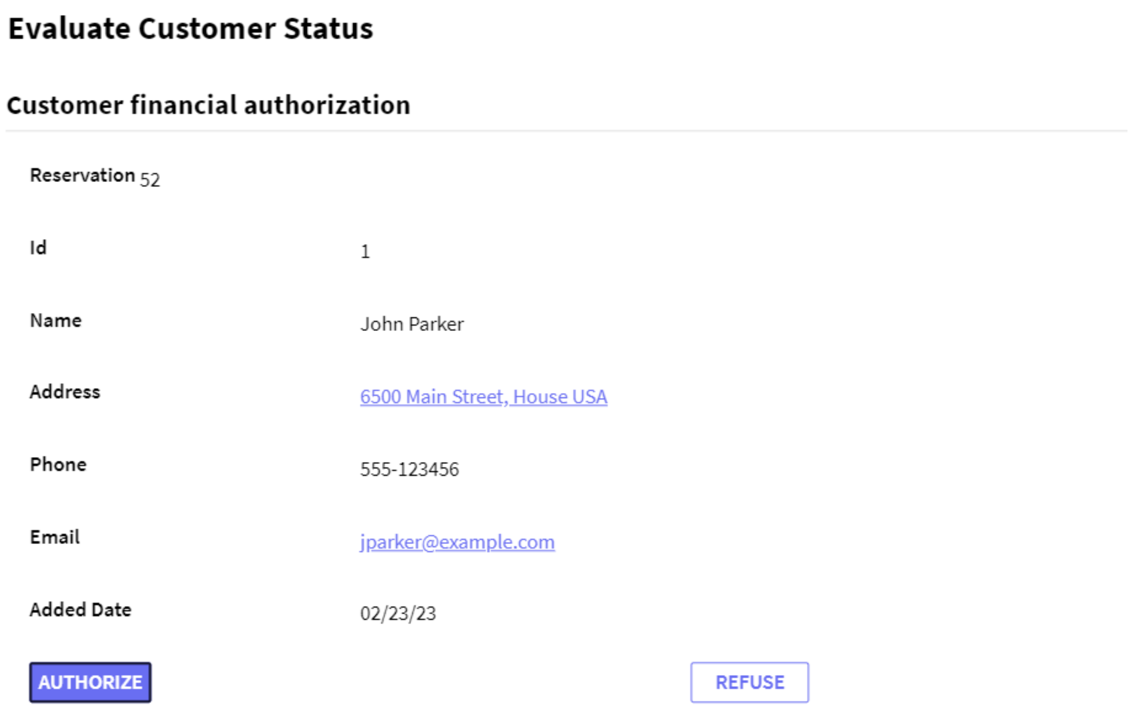
Size
32 KB
1,193 KB
9 KB
1,166 KB
8 KB
13 KB
45 KB
24 KB
10 KB
20 KB
17 KB
1,836 KB
14 KB
535 KB
35 KB
11 KB
45 KB
323 KB
104 KB
475 KB

In addition to the **Timer Scheduler** utility, we have a **Timer Control** that is run only once. The timer scheduler is typically used during the prototyping stage, and the timer control is used when the workflow is in production, with the operating system features to schedule its execution.

We go back to GeneXus, run the diagram **FlightTicketReservation** and move on to the screen that we were in before.

In the task "**Add customer information required for traveling**" we click on Send. Note that we don't move on to the next step in the diagram because the "Evaluate Customer" task is pending.

We execute it and see that the web panel screen is opened to evaluate the customer.



Evaluate Customer Status

Customer financial authorization

Reservation 52

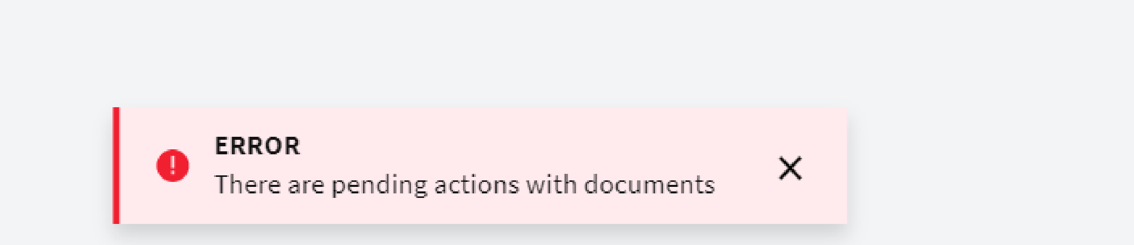
Id	1
Name	John Parker
Address	6500 Main Street, House USA
Phone	555-123456
Email	jparker@example.com
Added Date	02/23/23

We click on the **Authorize** button and complete the "**Evaluate Customer**" task. As we can see, the tasks "**Check reservation documents**" and "**Authorize reservation – Customer Care Manager**" are now pending.

The "Check reservation documents" task also has associated documents because we had set the Read operation of the visa as a required action.

We run the task and see that the work with documents window is opened. Instead of clicking on the "Read" button, we close the window and click on Send on the task.

An error message is displayed to indicate that documents still have to be read.

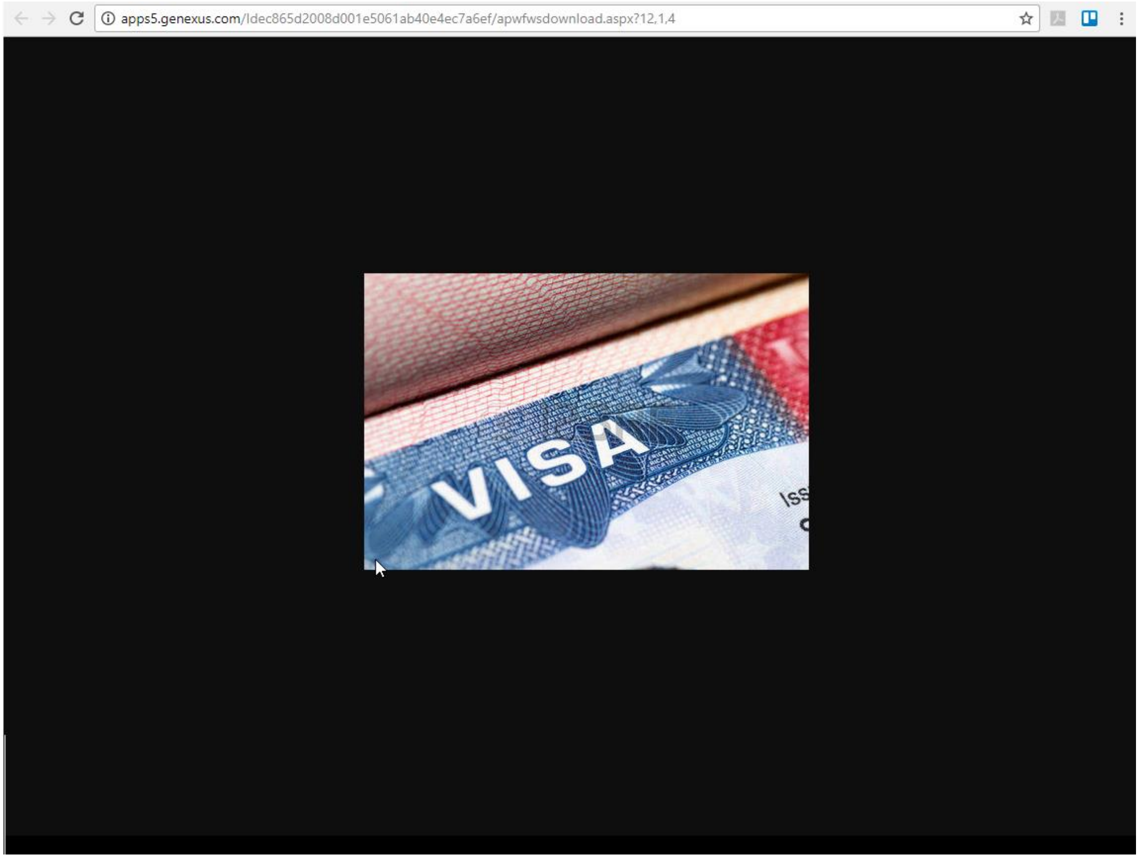


ERROR

There are pending actions with documents

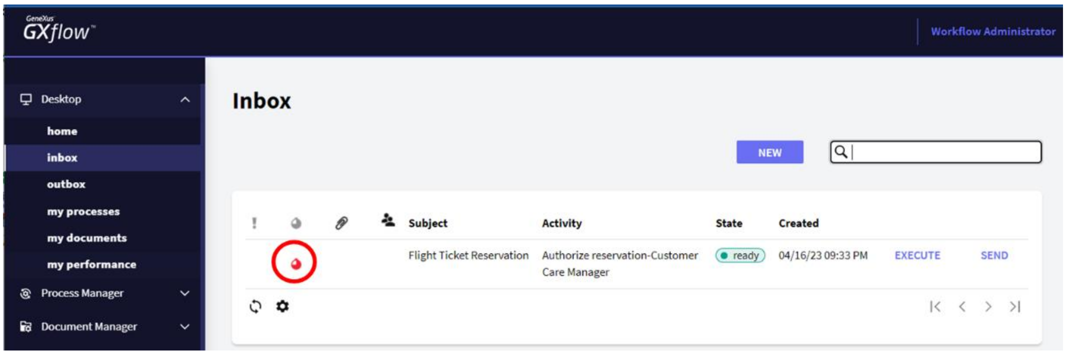
We run the task again, select the Visa document and click on Read.

A browser window is opened to show the document, which in this case contains an image of the visa.



apps5.genexus.com/ldec865d2008d001e5061ab40e4ec7a6ef/apwfwsdownload.aspx?12,1,4

We close the window, complete the task and see that it has been removed from the inbox.



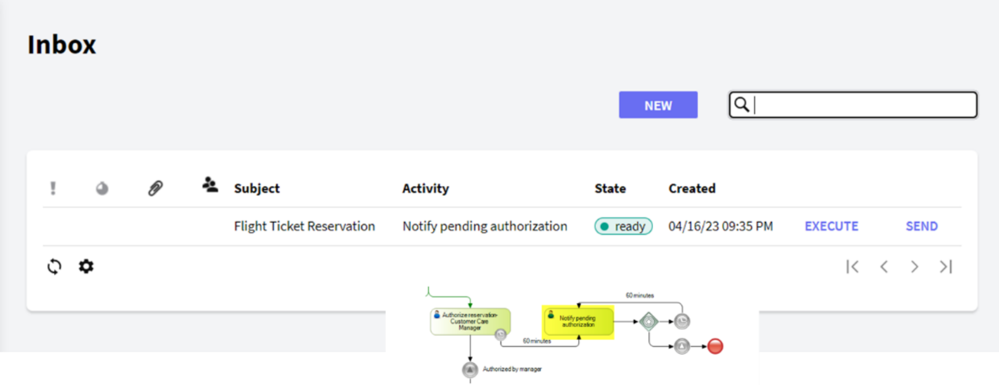
Inbox

!				Subject	Activity	State	Created		
				Flight Ticket Reservation	Authorize reservation-Customer Care Manager	ready	04/16/23 09:33 PM	EXECUTE	SEND

Instead of running the task “**Authorize reservation-Customer Care Manager**” we will wait a few minutes.

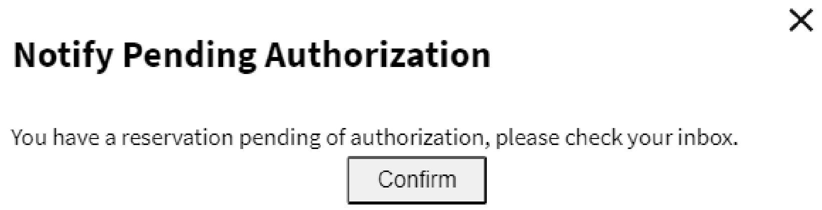
Note that the task has a clock icon.

We click on the Refresh button of the browser.


Inbox
NEW
Authorize reservation- Customer Care Manager
Notify pending authorization
60 minutes
60 minutes
Authorized by manager

!				Subject	Activity	State	Created		
				Flight Ticket Reservation	Notify pending authorization	● ready	04/16/23 09:35 PM	EXECUTE	SEND

Two minutes later, the interactive task **Notify Pending Authorization** is displayed on the inbox; this is the notification triggered by the timer. Let’s run it to show the notification that informs us about a pending authorization.



Notify Pending Authorization

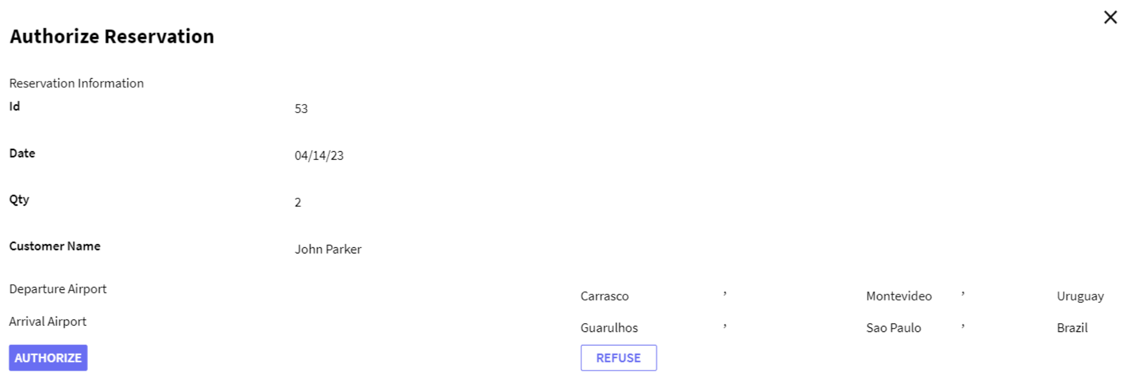
You have a reservation pending of authorization, please check your inbox.

We click on Confirm to close the window, and click on Send to complete the task.

These notifications will be shown every 2 minutes until the manager carries out the task.

Let’s run the notification task again…

Now let’s run the authorization task and click on Authorize.



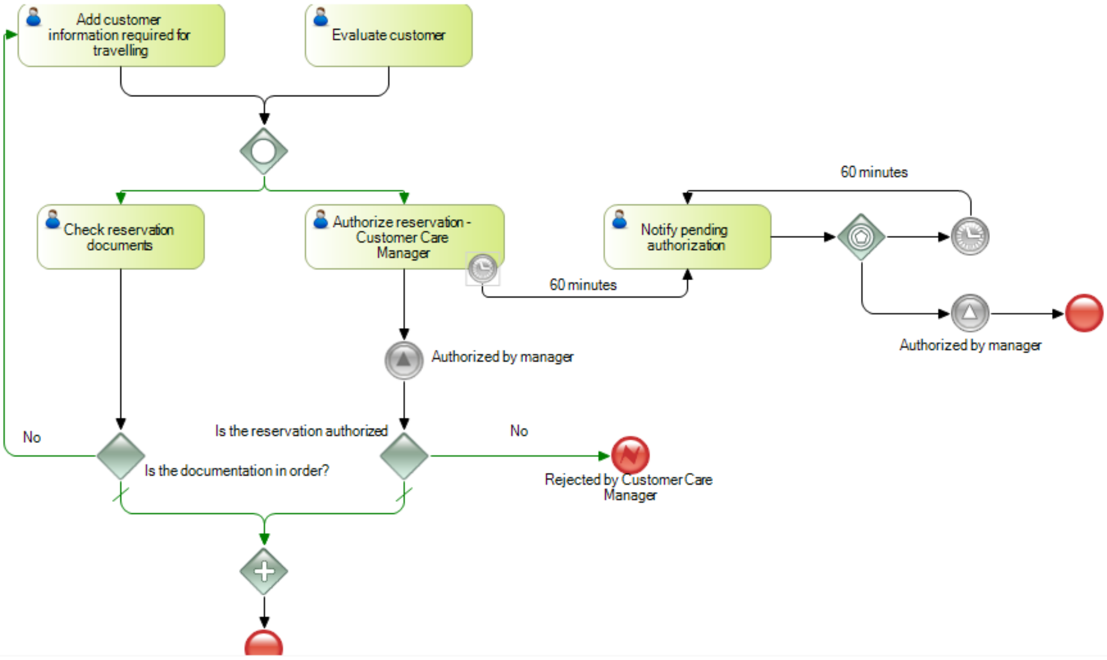
Authorize Reservation

Reservation Information

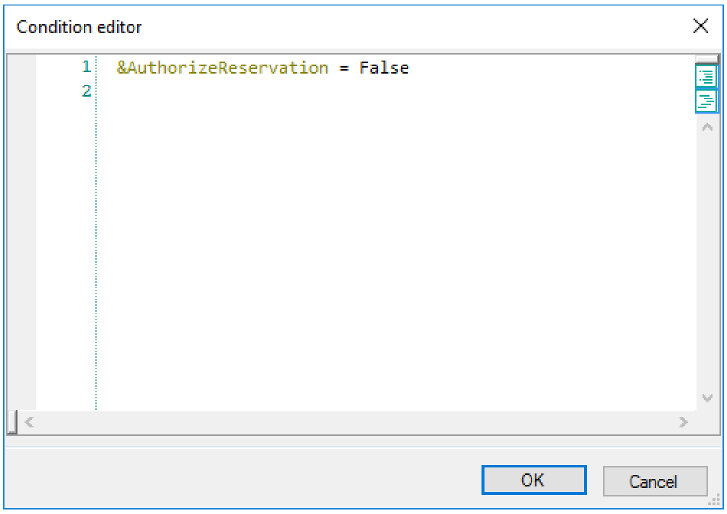
Id	53			
Date	04/14/23			
Qty	2			
Customer Name	John Parker			
Departure Airport		Carrasco ,	Montevideo ,	Uruguay
Arrival Airport		Guarulhos ,	Sao Paulo ,	Brazil

We complete the task and wait a few minutes…

The regular notifications are no longer displayed because the signal event placed after the task sent a message to the other signal event and cancelled this notification loop.

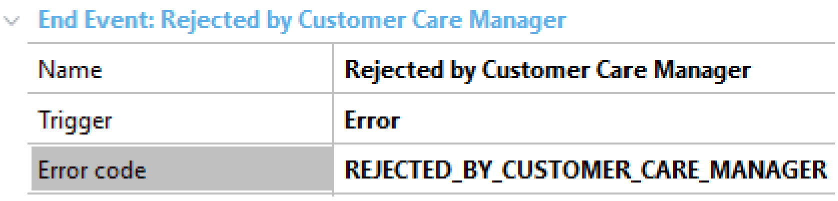

Add customer information required for travelling
Evaluate customer
60 minutes
Check reservation documents
Authorize reservation - Customer Care Manager
Notify pending authorization
60 minutes
Authorized by manager
Authorized by manager
No
Is the reservation authorized
No
Is the documentation in order?
Rejected by Customer Care Manager

We go back to the diagram and see that the signal event "Authorized by manager" goes to an Exclusive Gateway that evaluates if the reservation was authorized or not. To set this decision, we double-click on the connector that joins the Gateway with the Error End Event "Rejected by Customer Care Manager" and in the condition editor we type &AuthorizeReservation = False



```
&AuthorizeReservation = False
```

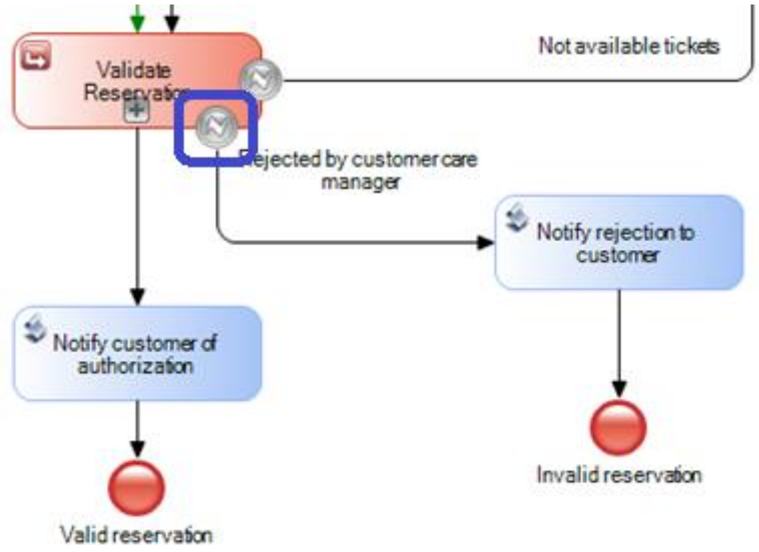
This error event will inform the FlightTicketReservation process that the reservation validation process has been canceled because the customer care manager didn't authorize the reservation. In its properties, we look at the error code that it has assigned.



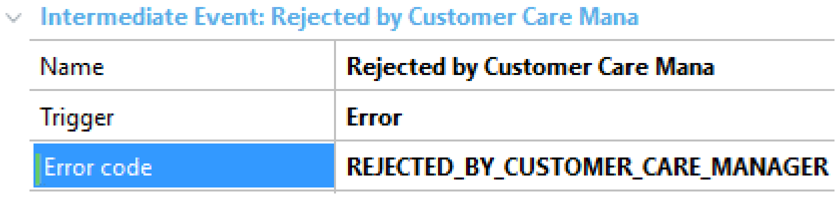
End Event: Rejected by Customer Care Manager

Name	Rejected by Customer Care Manager
Trigger	Error
Error code	REJECTED_BY_CUSTOMER_CARE_MANAGER

In the FlightTicketReservation process we have a Signal Error Event that is set as catch and has the same name.


Validate Reservation
Not available tickets
Rejected by customer care manager
Notify rejection to customer
Notify customer of authorization
Invalid reservation
Valid reservation

Most importantly, it has the same Error Code, which makes sure that it will capture the event triggered in the ValidateReservation subprocess. As a result, a notification message will be sent to the customer.



Intermediate Event: Rejected by Customer Care Mana

Name	Rejected by Customer Care Mana
Trigger	Error
Error code	REJECTED_BY_CUSTOMER_CARE_MANAGER

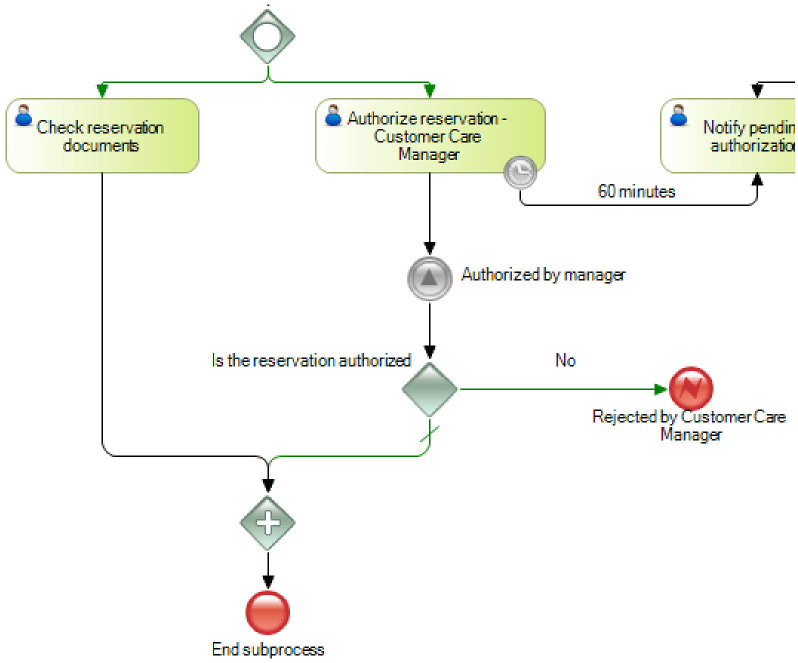
We go back to the reservation validation subprocess and see that the document revision task also ends in an Exclusive Gateway. In this case, it evaluates whether the documentation is in order; otherwise, the task "**Add customer information required for traveling**" will have to be carried out again.

When we created the model, we assumed that the interactive task "**Check reservation documents**" would be associated with a GeneXus object that would load a relevant data item with a value that would later be evaluated in the Exclusive Gateway.

However, we've seen that the same workflow engine checks that we read the document, since we had created a Read operation and set it as Required.

Therefore, we don't need a Gateway to check whether the task has been carried out or not, because the control is made by the engine itself when we run the task. We are not allowed to continue if we don't Read this document.

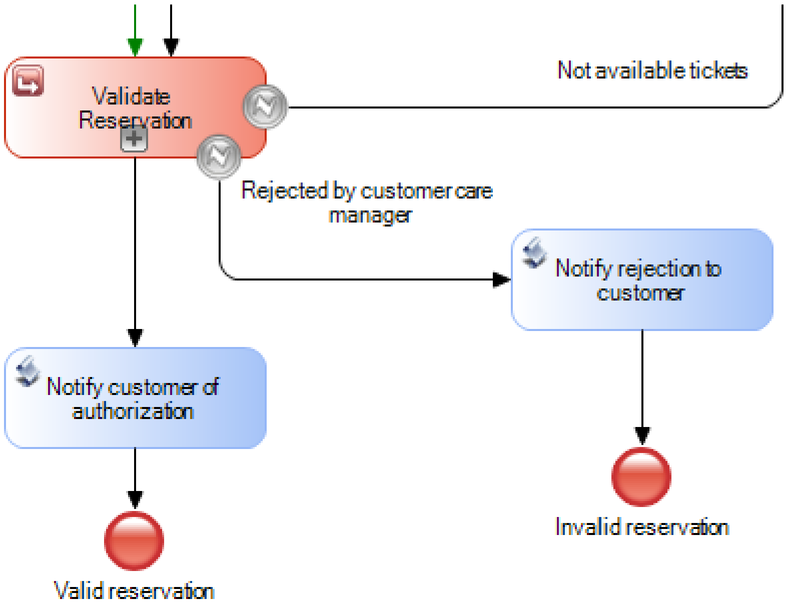
We change the diagram by deleting the Exclusive Gateway "**Is the documentation in order?**"; as a result, the task "**Check reservation documents**" is now connected directly to the Parallel Gateway.


Check reservation documents
Authorize reservation - Customer Care Manager
Notify pendin authorizatio
60 minutes
Authorized by manager
Is the reservation authorized
No
Rejected by Customer Care Manager
End subprocess

The Parallel Gateway synchronizes all the paths connected to it in the diagram and waits until all the paths arrive to it to allow the process to continue.

When we run the document control task and review the documents, if the reservation was authorized by the customer care manager, the Parallel Gateway will allow the flow to continue and the subprocess will end.

In this case, the validation ended successfully; therefore, the control will return to the FlightTicketReservation main process.


Validate Reservation
Not available tickets
Rejected by customer care manager
Notify rejection to customer
Notify customer of authorization
Invalid reservation
Valid reservation

Upon returning to this process with a successful validation, the customer will be notified that the reservation was authorized and the ticket reservation process will end.

In the next videos we will make changes to the diagram to meet new requirements made by the travel agency.