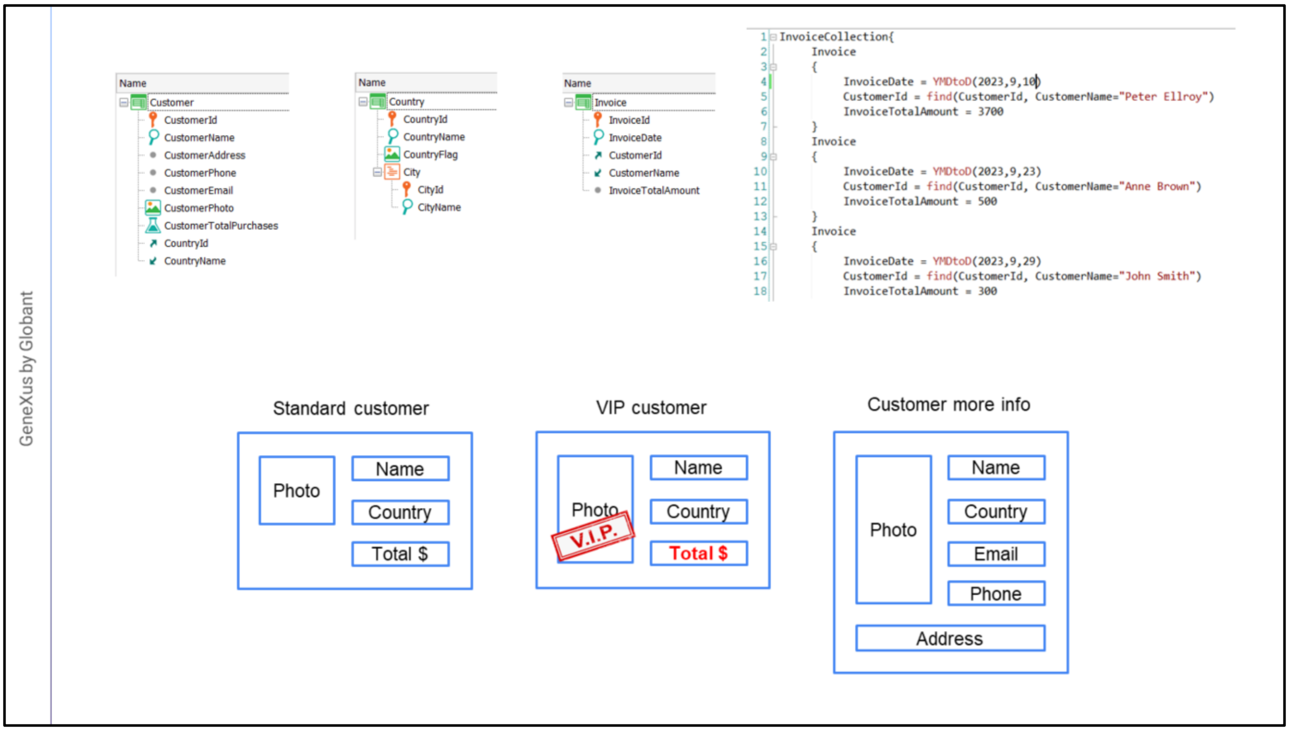
GeneXus by Globant

Multiple Layouts in a Grid

Rodolfo Roballo

We have seen that it is possible to assign different layouts to a panel. In this video, we will see that it is also possible to assign different layouts to the rows of a grid so that their appearance varies depending on their content, or to change the design when tapping on a particular row.



Name
- Customer
  - CustomerId
  - CustomerName
  - CustomerAddress
  - CustomerPhone
  - CustomerEmail
  - CustomerPhoto
  - CustomerTotalPurchases
  - CountryId
  - CountryName

Name
- Country
  - CountryId
  - CountryName
  - CountryFlag
  - City
    - CityId
    - CityName

Name
- Invoice
  - InvoiceId
  - InvoiceDate
  - CustomerId
  - CustomerName
  - InvoiceTotalAmount

```
InvoiceCollection{
	Invoice
	{
		InvoiceDate = YMDtoD(2023,9,10)
		CustomerId = find(CustomerId, CustomerName="Peter Ellroy")
		InvoiceTotalAmount = 3700
	}
	Invoice
	{
		InvoiceDate = YMDtoD(2023,9,23)
		CustomerId = find(CustomerId, CustomerName="Anne Brown")
		InvoiceTotalAmount = 500
	}
	Invoice
	{
		InvoiceDate = YMDtoD(2023,9,29)
		CustomerId = find(CustomerId, CustomerName="John Smith")
		InvoiceTotalAmount = 300
```

We will start from a transaction model with one customer transaction, one country transaction, and one invoice transaction. The invoice transaction has no rows, because at this point we only care about the invoice total and the details of the customer on the invoice. We have also created the Invoice_DataProvider with the invoice data for each customer.
The total invoiced to each customer is calculated as the sum of their invoices, using the formula attribute CustomerTotalPurchases.
If you want to have these objects, import the file invoices.xpz found in the Materials section of the course page.

We want to see the list of customers, with the customer's name, photo, country, and total invoiced so far. In addition, for those customers whose invoice total exceeds $5000, the total should be displayed in bold and in a different color, with an image over the photo to indicate that it is a VIP customer, while the rest of the customers should be displayed without this highlighting.

That is, we are going to vary the content and appearance of the information in each row, depending on the information displayed. Moreover, when we tap on any customer, we want to expand their information, also showing their email, phone, and address but without the total purchases.

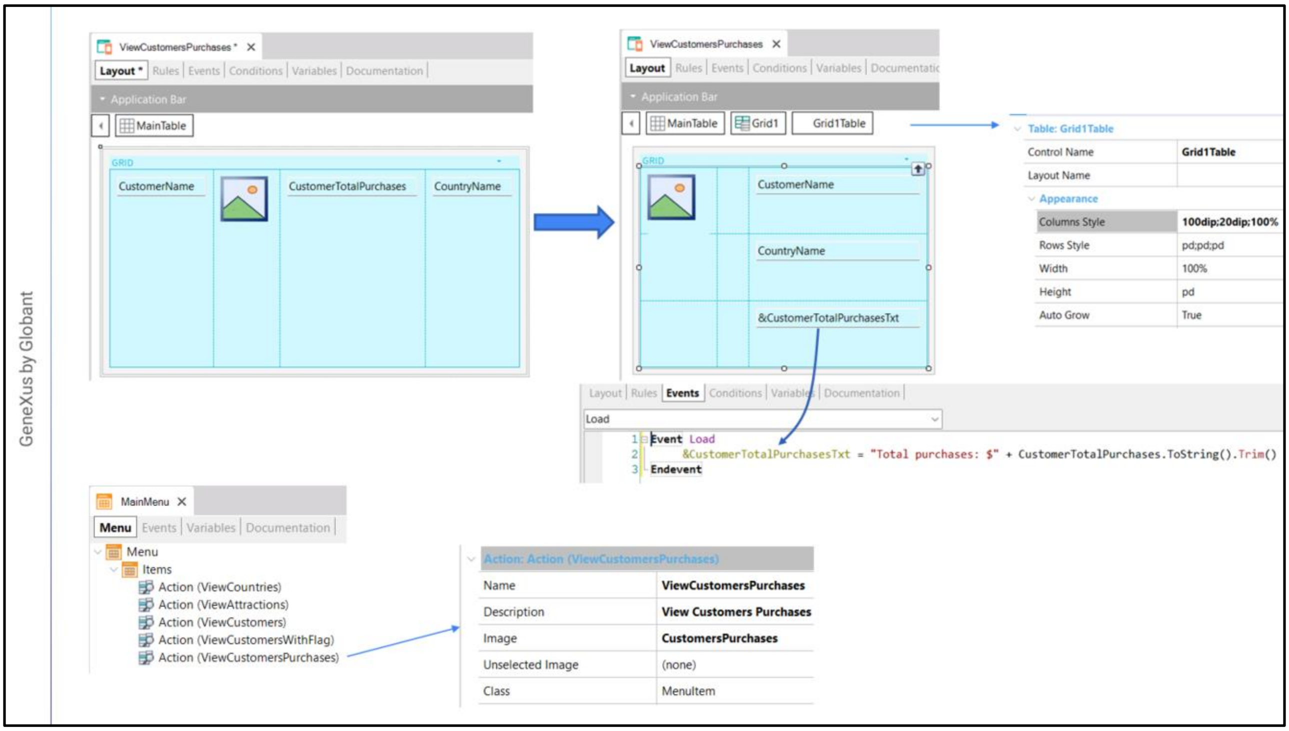


Table: Grid1Table

Control Name	Grid1Table
Layout Name	
Appearance	
Columns Style	100dip;20dip;100%
Rows Style	pd;pd;pd
Width	100%
Height	pd
Auto Grow	True

Layout | Rules | Events | Conditions | Variables | Documentation

Load

```
Event Load
	&CustomerTotalPurchasesTxt = "Total purchases: $" + CustomerTotalPurchases.ToString().Trim()
Endevent
```

- Menu
  - Items
    - Action (ViewCountries)
    - Action (ViewAttractions)
    - Action (ViewCustomers)
    - Action (ViewCustomersWithFlag)
    - Action (ViewCustomersPurchases)

Action: Action (ViewCustomersPurchases)

Name	ViewCustomersPurchases
Description	View Customers Purchases
Image	CustomersPurchases
Unselected Image	(none)
Class	MenuItem

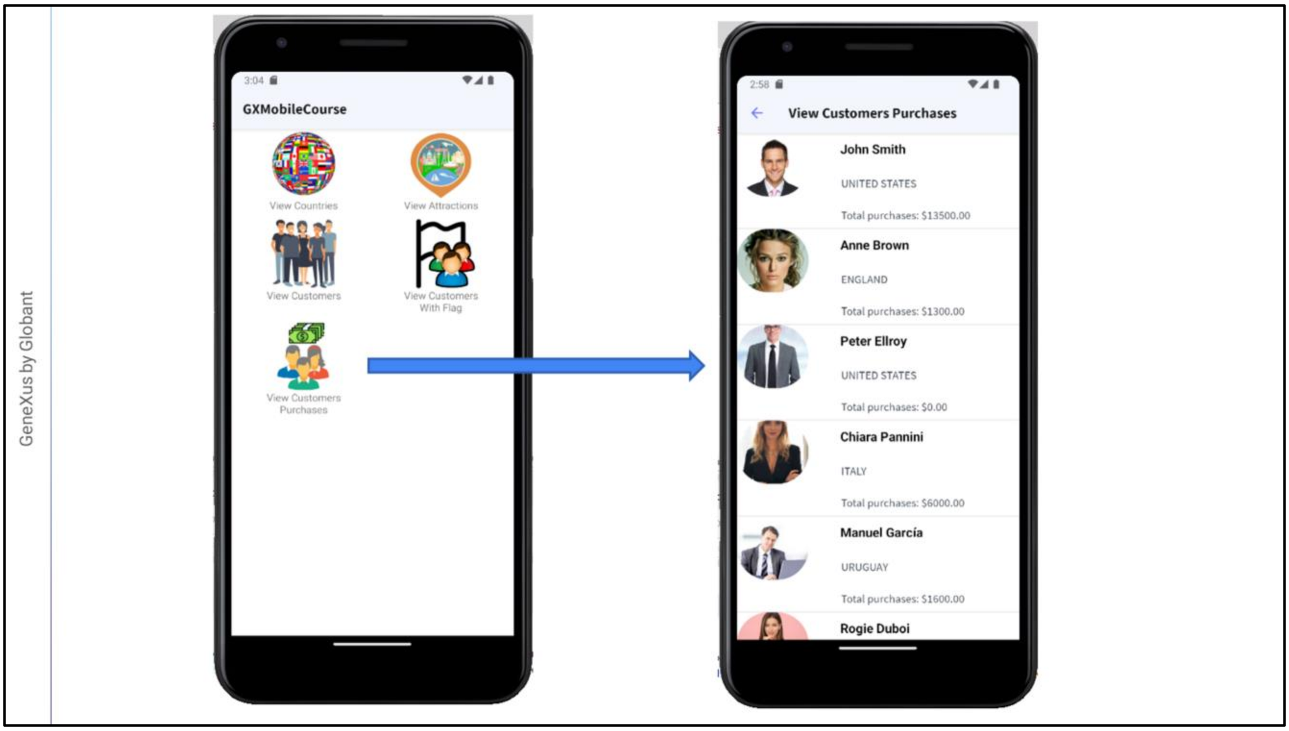
We start by creating a ViewCustomersPurchases panel and dragging a grid control with the attributes CustomerName, CustomerPhoto and CountryName. We also create a &CustomerTotalPurchasesTxt variable of Varchar(40) type and add it to the grid.

We move the attributes to achieve the layout we had planned, with the photo on the left and the rest of the data on the right. We add a column in the middle as a separator. To the attributes on the right we set Horizontal Alignment to Left and Vertical Alignment to Middle. Now we distribute the column spacing by setting the Columns Style property in the Grid1Table table with the values: 100 dip, 20 dip, and 100%.

As for the CustomerPhoto attribute, we set its Row Span property to the value 2 so that it takes 2 rows high. We also assign the ImageRound class that we had previously defined in the Design System created. And in the CustomerName attribute we set the Class property to the AttributeBold class.

In the Events tab, we select the Load event and assign the variable &CustomerTotalPurchasesTxt with the text "Total purchases: $“, and concatenate it to the CustomerTotalPurchases attribute converted to string and without spaces.

We open the MainMenu and drag the ViewCustomersPurchases panel to the Items node. Next, in the action that was created, we assign the CustomersPurchases image to the Image property. We run Android Studio and open the emulator that we had created. We press F5.



View Customers Purchases

John Smith
UNITED STATES
Total purchases: $13500.00

Anne Brown
ENGLAND
Total purchases: $1300.00

Peter Ellroy
UNITED STATES
Total purchases: $0.00

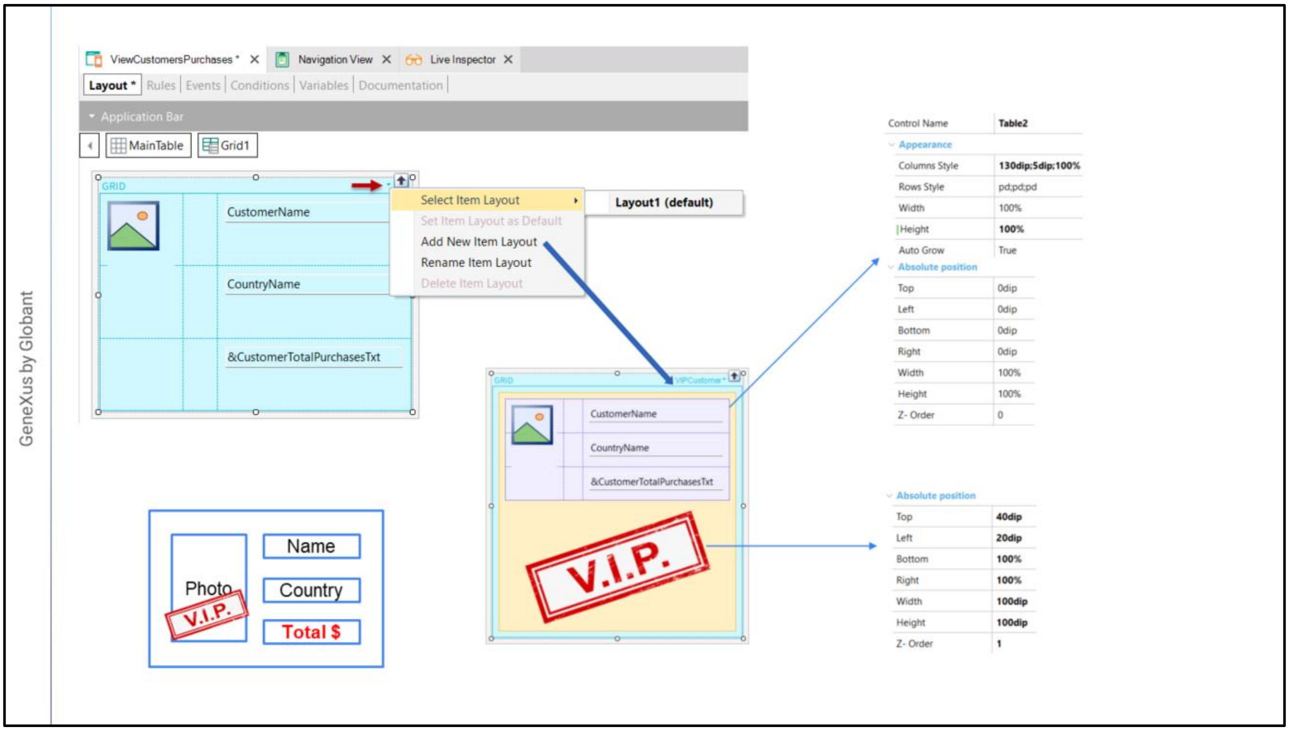
Chiara Pannini
ITALY
Total purchases: $6000.00

Manuel García
URUGUAY
Total purchases: $1600.00

Rogie Duboi

We tap on the View Customers Purchases icon and see the list of customers, with their personal data and invoiced amounts, as we expected.

Now we are going to meet the first requirement, which implied that customers who exceeded $5000 should be highlighted with a different design than customers who had not reached that amount.



Control Name	Table2
Appearance	
Columns Style	130dip;5dip;100%
Rows Style	pd;pd;pd
Width	100%
Height	100%
Auto Grow	True
Absolute position	
Top	0dip
Left	0dip
Bottom	0dip
Right	0dip
Width	100%
Height	100%
Z- Order	0

Absolute position	
Top	40dip
Left	20dip
Bottom	100%
Right	100%
Width	100dip
Height	100dip
Z- Order	1

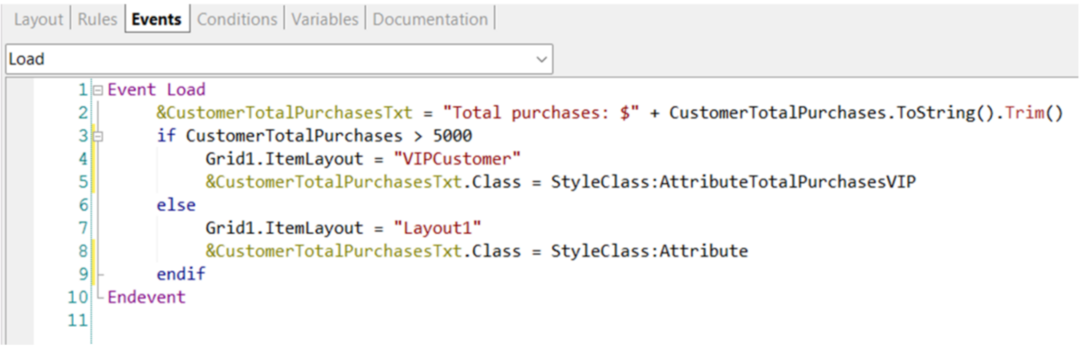
To have each row’s design change according to what we need, we must create a new layout for the grid.

If we click on the little green arrow at the top right of the grid, a menu opens indicating that the default layout selected is Layout1 and that we can rename it or create a new layout.

Let's select Add New Item Layout and name it VIPCustomer. The name appears in the upper right corner and the new layout is initialized with the design of the default layout, but we can make the changes we want and both layouts will be completely independent.

To place the VIP stamp on the photo, we need a Canvas control, so we start by removing all the controls from the grid... we drag a Canvas control to the grid and then we drag a table to the canvas, where we put back the attributes with the design we had. In the table, we are going to adjust the Column Style values by setting the photo to 130dip, the separator column to 5 dip, and the third column –the data column– to 100%. The Absolute position properties are left by default, so the Z-Order will be set to 0 and this grid will be in the bottom layer of the layout.

We drag an image below the table and associate it with the VIPCustomer image; an image with transparent background, which we had in the KB. In the Absolute position section properties, we set Top to 40dip, Left to 20dip, Bottom and Right to 100% and Width and Height to 100dip, and Z-Order to 1 so that the VIP stamp is on the customer's photo.



```
Event Load
	&CustomerTotalPurchasesTxt = "Total purchases: $" + CustomerTotalPurchases.ToString().Trim()
	if CustomerTotalPurchases > 5000
		Grid1.ItemLayout = "VIPCustomer"
		&CustomerTotalPurchasesTxt.Class = StyleClass:AttributeTotalPurchasesVIP
	else
		Grid1.ItemLayout = "Layout1"
		&CustomerTotalPurchasesTxt.Class = StyleClass:Attribute
	endif
Endevent
```

In the events section, we add the following code so that if the customer's total purchases exceed $5000, we assign the VIPCustomer layout to the grid and otherwise we assign Layout1.

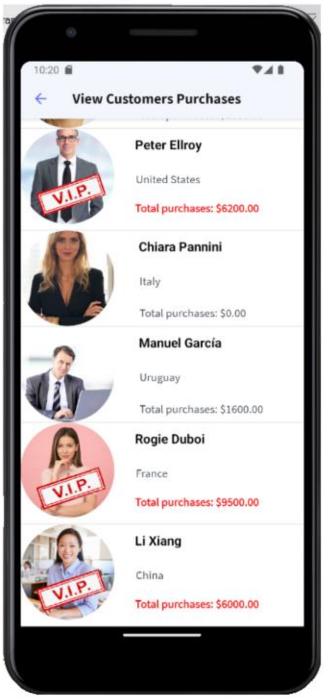
We will also take advantage of the logical branching to assign a class to a control at runtime, so that the format of the total purchases also depends on whether the total purchases exceed the requirement for the VIP customer or not.

First, we open the GXMobileCourseDSO Design System object and create an AttributeTotalPurchasesVIP class that inherits from AttributeBold, and define a font size of 16 pixels and red color. We save.

Now we return to the panel events and where the customer is VIP, we write the &CustomerTotalPurchasesTxt variable, period, Class, and assign the AttributeTotalPurchasesVIP class using StyleClass. In else, we assign the Attribute class.

In this way, the total purchases will be highlighted in red only for VIP customers.

We press F5...



10:20

View Customers Purchases

Peter Ellroy
United States
Total purchases: $6200.00

Chiara Pannini
Italy
Total purchases: $0.00

Manuel García
Uruguay
Total purchases: $1600.00

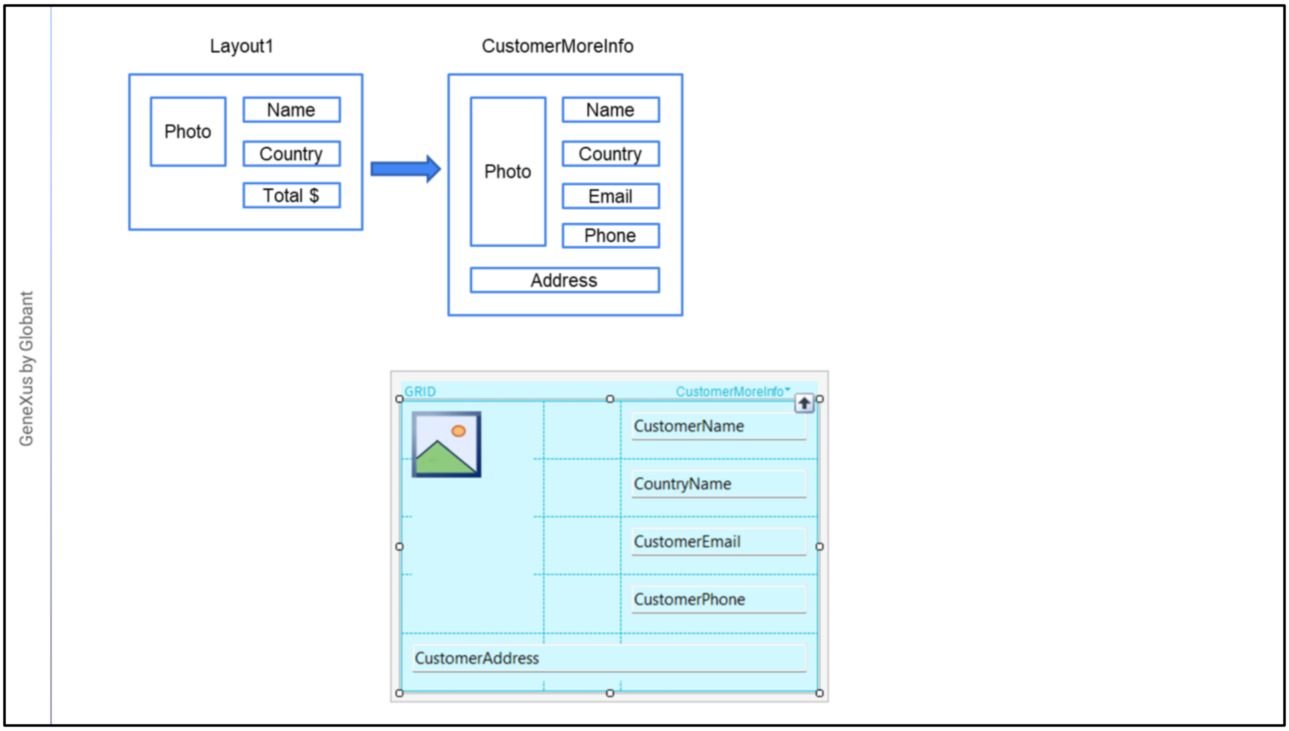
Rogie Duboi
France
Total purchases: $9500.00

Li Xiang
China
Total purchases: $6000.00

We tap on the View Customers Invoices icon and see that some customers have the VIP customer stamp and others do not. We check that the customers who have the stamp are those whose purchases exceed $5000 in total as we wanted, and that the total for these customers is highlighted in red.

For example, Peter Ellroy reached 6200 and is VIP, and if we go further down we see that Rogie and Li are also VIP customers.

For the other customers who didn't reach this total, this data is not highlighted, meaning that based on a condition we were able to have some rows formatted differently than others and this formatting was changed at runtime depending on the data.



Layout1

Photo

Name

Country

Total $

CustomerMoreInfo

Photo

Name

Country

Email

Phone

Address

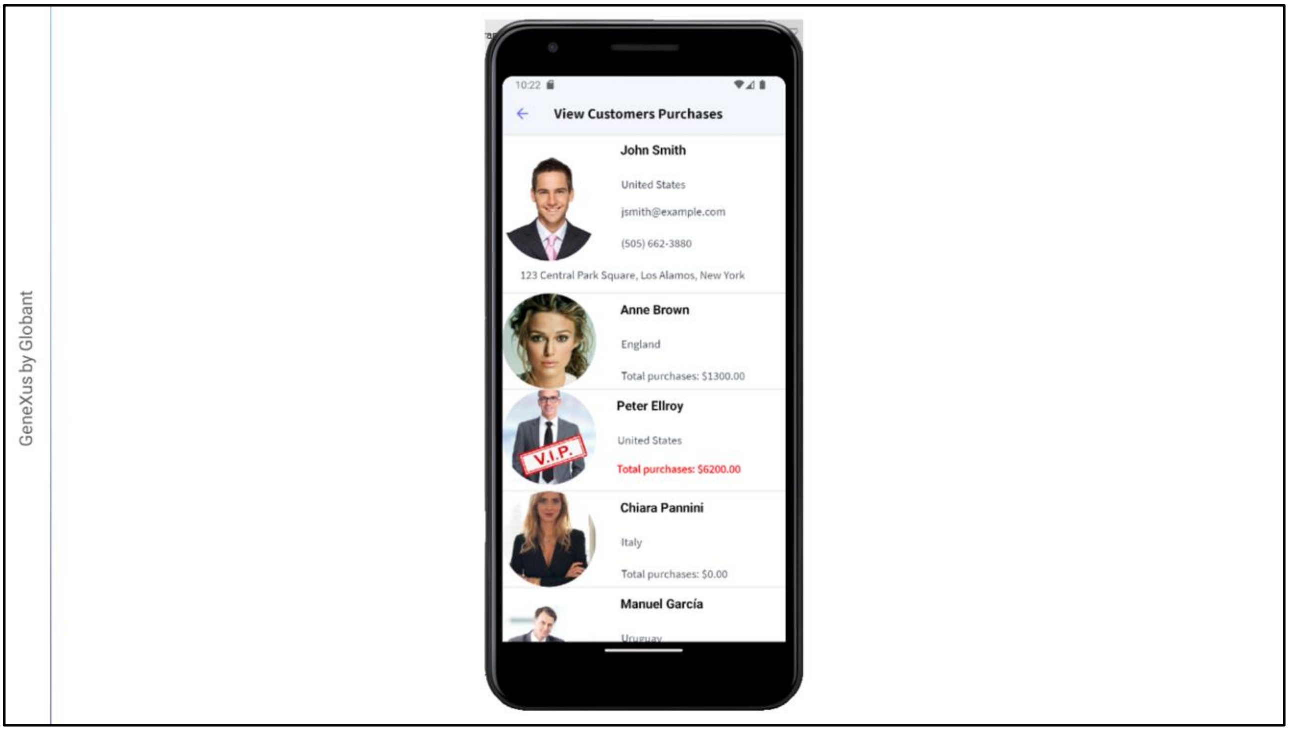
Now we will meet the last requirement, so that when we tap on a customer, their extended information is displayed.

When we tap on a row, we want the layout of the customer's data to change and to also show the customer's name and country, email, phone and address, but without showing the total invoices.

For this we have to create a new layout for the grid, based on Layout1 which is the default layout, so first we select this layout and then click on the green arrow, select Create new layout, and name it CustomerMoreInfo.

We delete the CustomerTotalPurchases attribute and add the attributes CustomerEmail, CustomerPhone and CustomerAddress... We adjust the panel layout... align all the fields with horizontal justification to the left and with center VerticalAlignment... In CustomerAddress, we set ColSpan to 3... and save.

To have this layout used when we tap on a row, we go to the grid properties and in the Default Select Item Layout property we select the CustomerMoreInfo layout. We press F5 again...



10:22

View Customers Purchases

John Smith
United States
jsmith@example.com
(505) 662-3880
123 Central Park Square, Los Alamos, New York

Anne Brown
England
Total purchases: $1300.00

Peter Ellroy
United States
Total purchases: $6200.00

Chiara Pannini
Italy
Total purchases: $0.00

Manuel García
Uruguay

We open View Customer Purchases and now we tap on the first customer. And we see that its layout has changed showing all the customer's personal data.

If we tap again we return to the default layout. We also try the same on a VIP customer and see that the complete customer data is also displayed.

In this video, we saw how to assign different layouts to a grid at design time and at runtime, achieving a richer user interface.

In the following videos, we will continue working with other screen controls that provide functionality and design to our application.