

## **FLows USED IN PROJECT :**

### **FLOW 1:**

#### **ATTENDEE REGISTRATION FORM :**

**Process Type:** Screen Flow

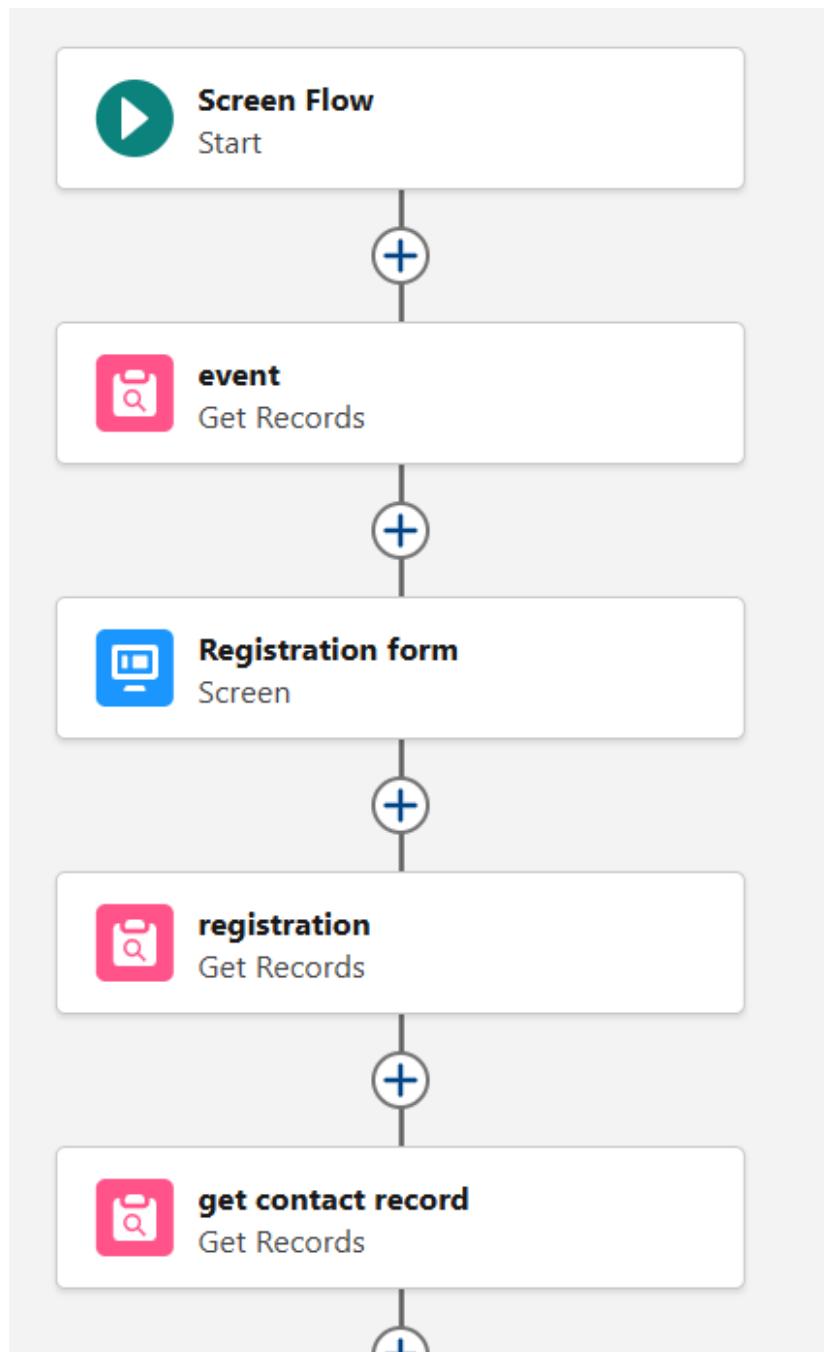
#### **DESCRIPTION:**

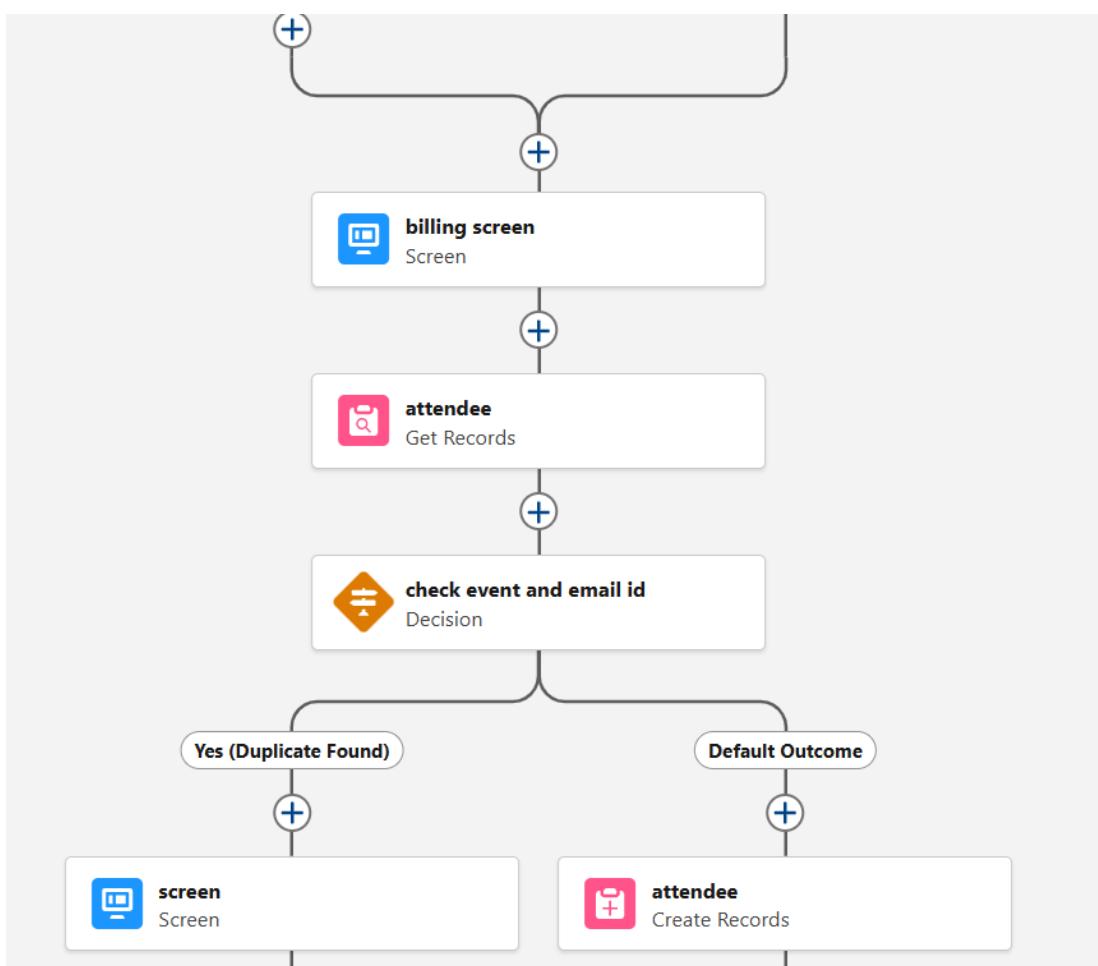
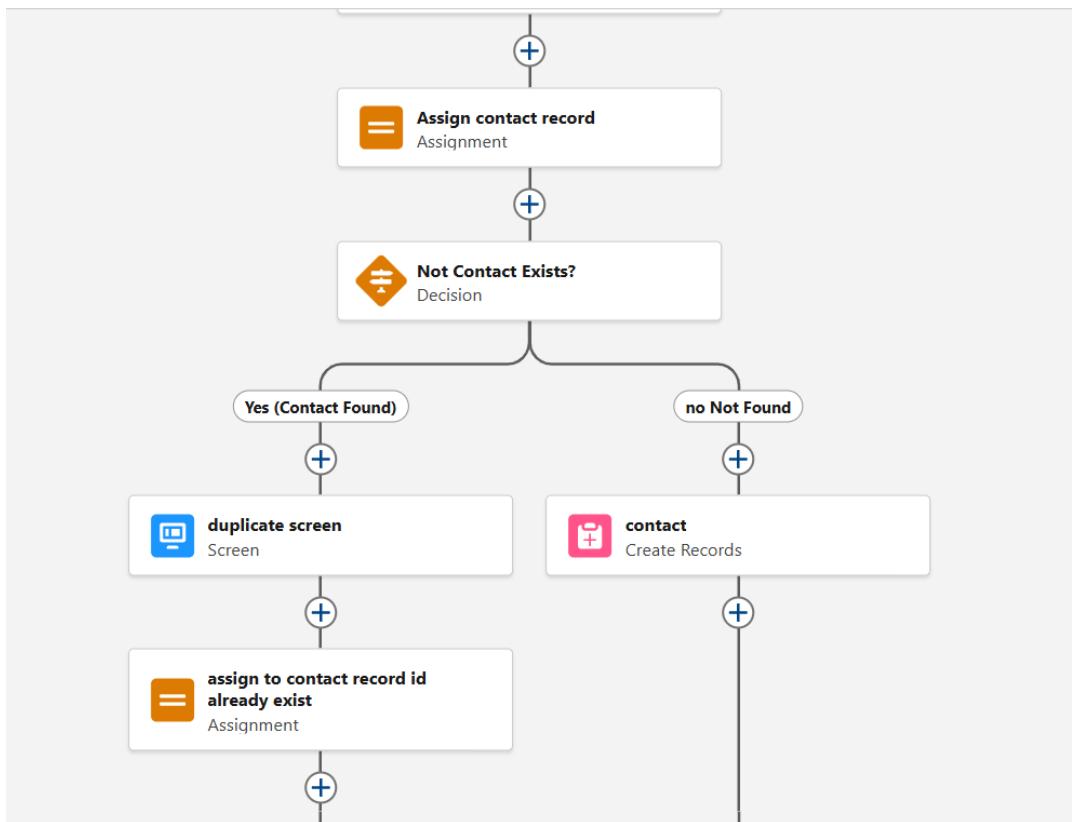
This is a Screen Flow.

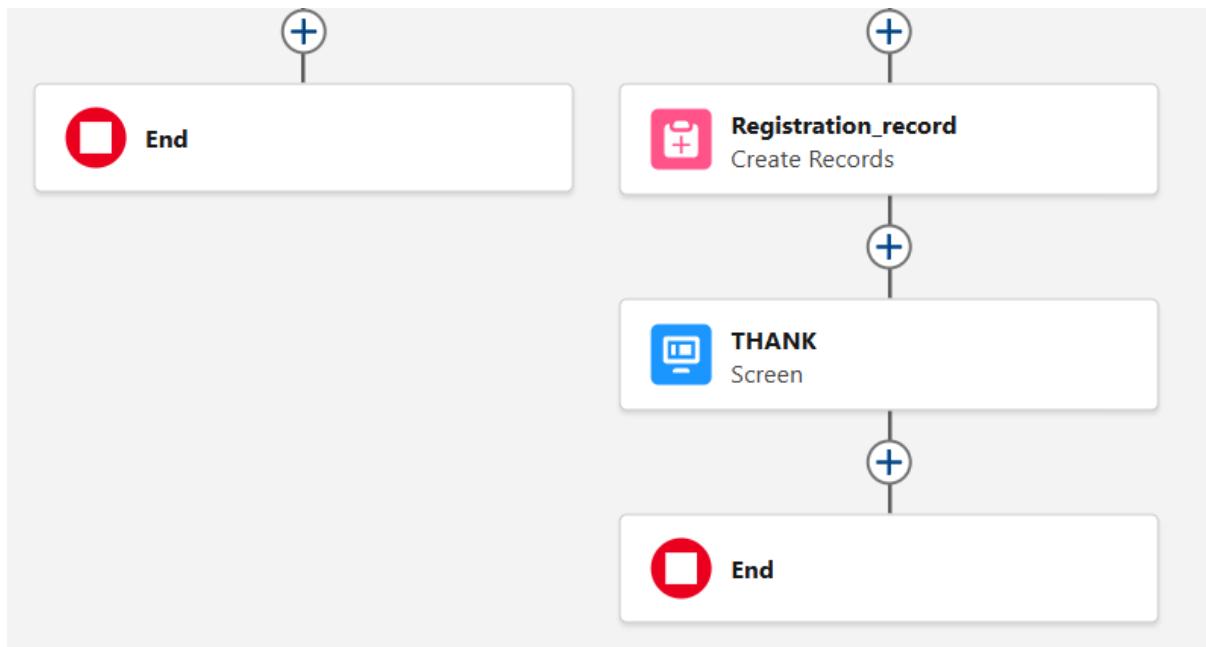
The flow begins by checking if the attendee is already registered for the selected event. If not, it proceeds to collect attendee details, validate inputs, and allow the user to proceed to the billing screen. If the attendee is already registered, the flow displays an error message and asks the user to re-enter their details.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Event\_c - Upcoming\_Events\_c : The flow filters events based on this field.
- \* Attendee\_List\_c - Email\_c : The flow uses this field to check if an attendee is already registered.
- \* Contact - Email : The flow creates a new contact record using this field.
- \* Registration\_c - Attendee\_c : The flow creates a new registration record using this field.







## FLOW 2:

### ATTENDEE REGISTRATION EMAIL FLOW:

**Process Type:** Autolaunched Flow

#### DESCRIPTION:

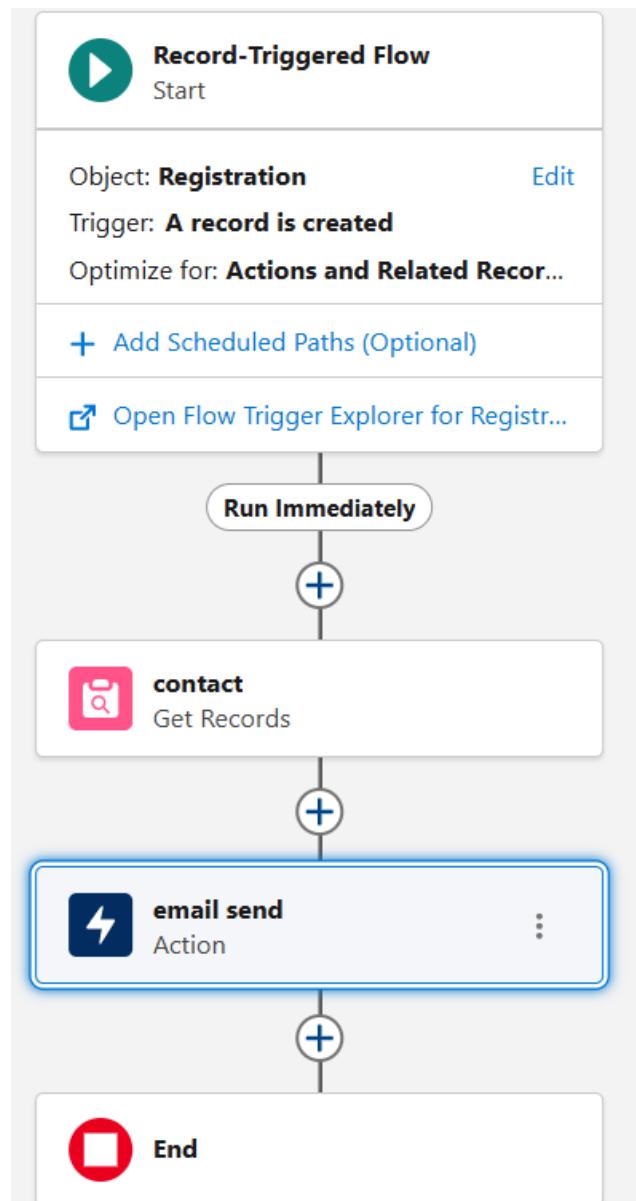
This is a Record-Triggered Flow.

This flow is triggered after a new 'Registration\_\_c' record is created. It checks if the associated contact has an email address and sends an email using a predefined template if the condition is met.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Registration\_\_c - Attendee\_\_r.Contact\_\_r.Id : The flow retrieves the ID of the associated contact from the 'Attendee\_\_r' lookup field on the 'Registration\_\_c' record.

- \* Contact - Email : The flow checks if the 'Email' field of the contact is not null.



## EMAIL TEMPLATE:

Dear {!Registration\_c.Attendee\_c} ,

Thank you for registering for {!Registration\_c.Event\_c}  
We're excited to have you join us.

Thank You

## **FLOW 3:**

### **SPEAKER REGISTRATION FLOW:**

**Process Type:** Screen Flow

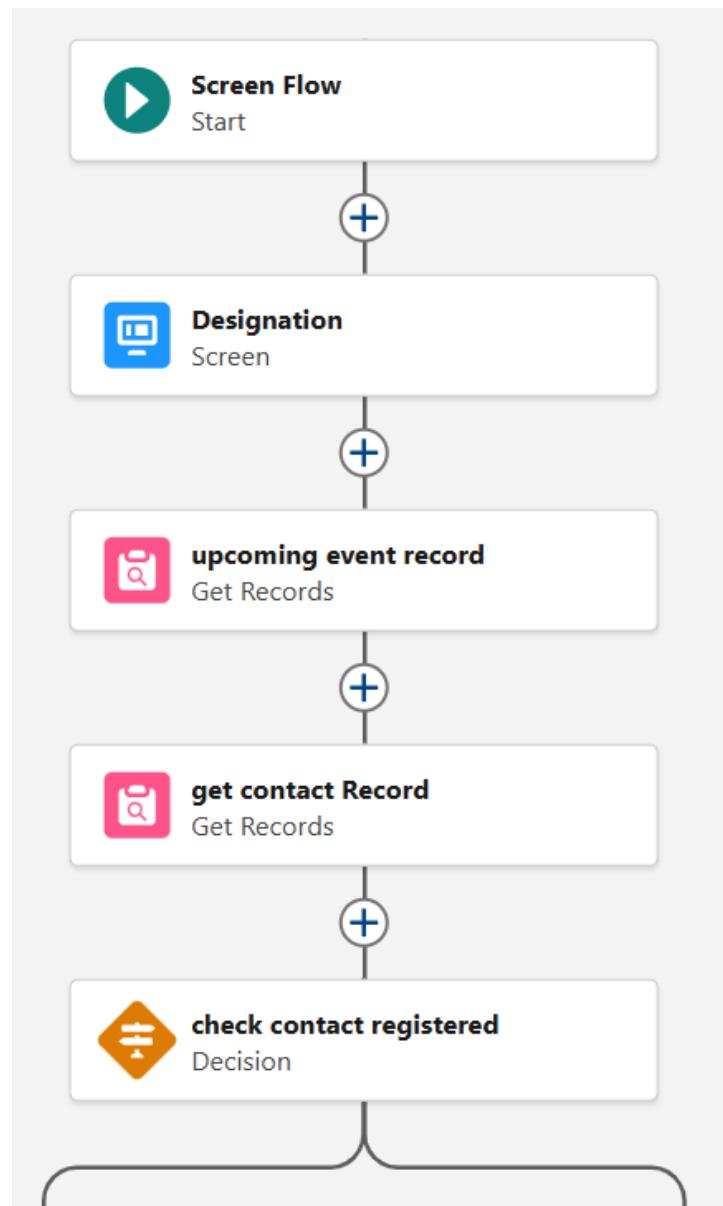
#### **DESCRIPTION:**

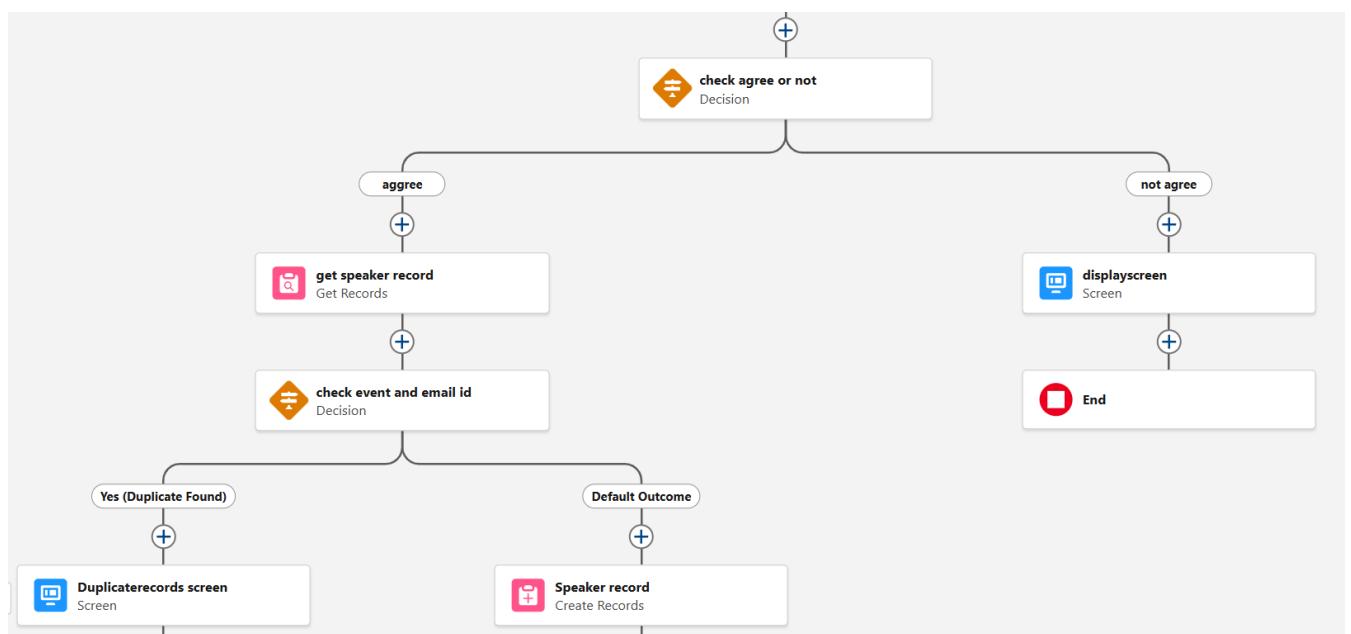
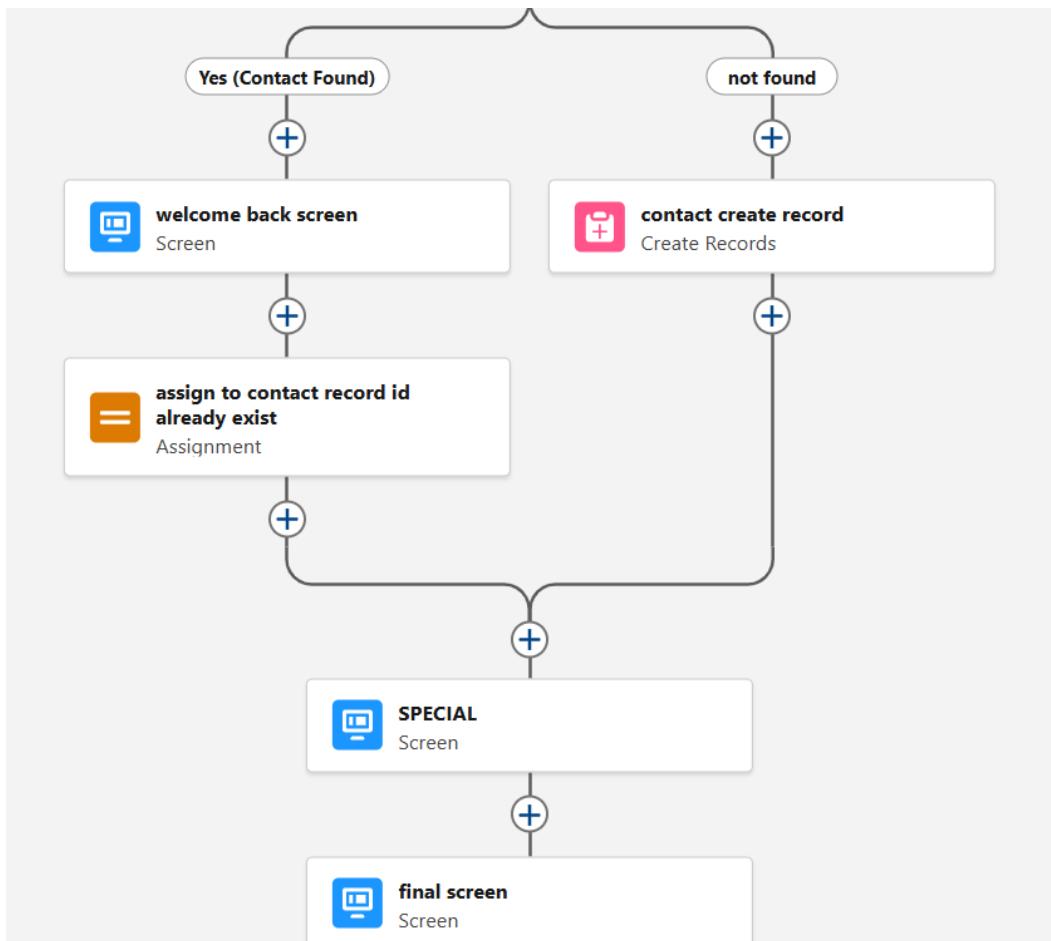
This is a Screen Flow.

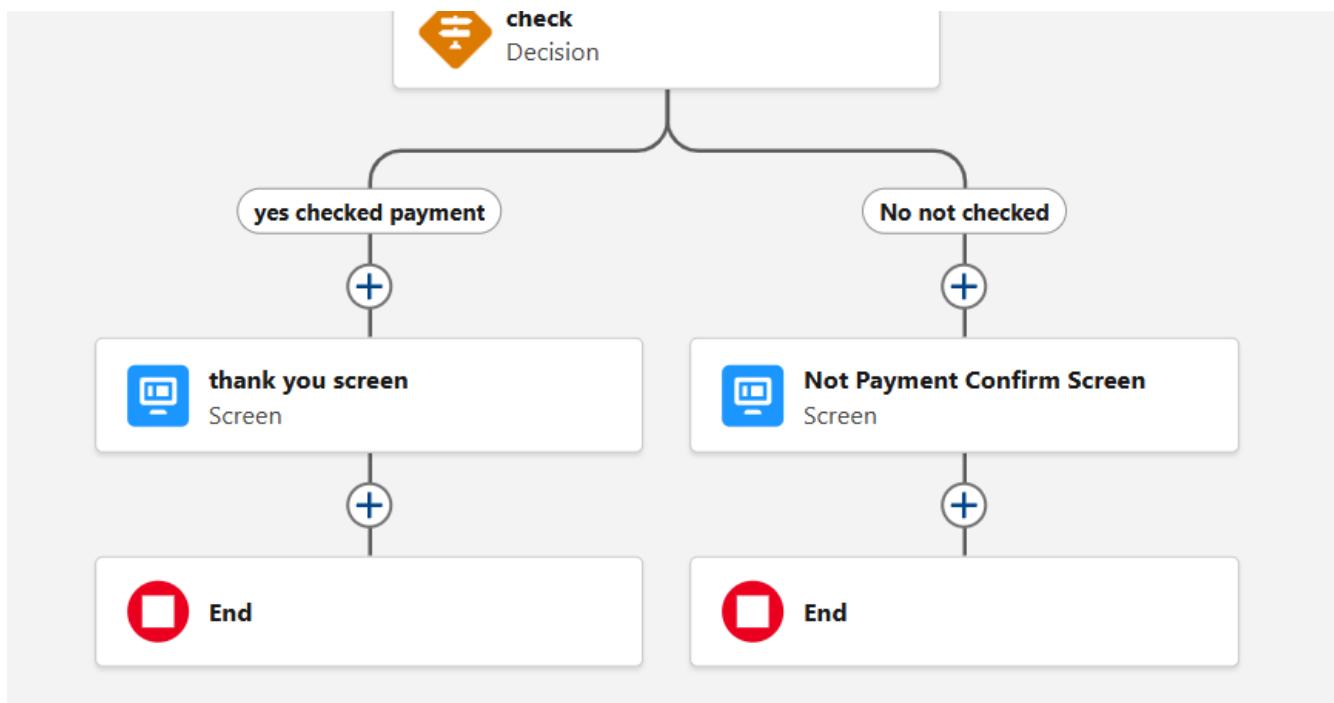
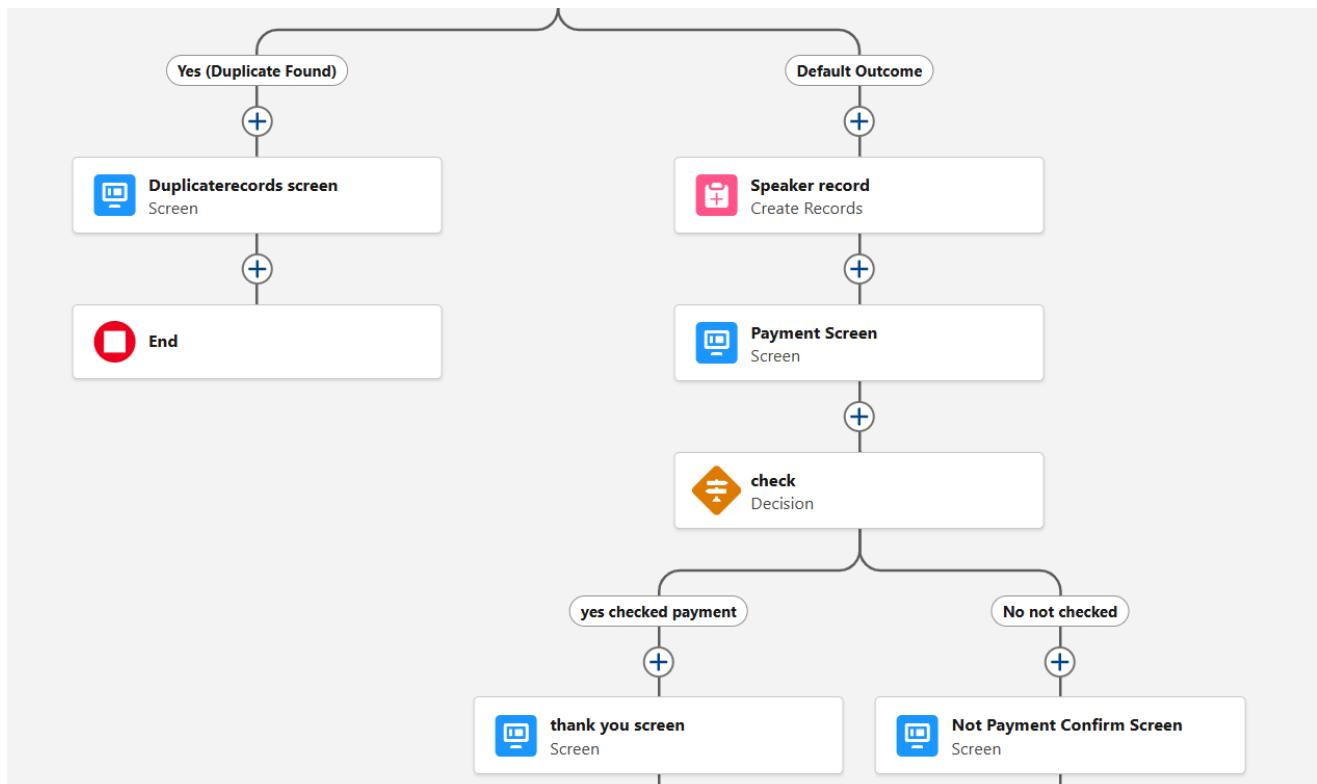
The flow begins with a screen where speakers enter their basic information, followed by screens for designation, topic, and one-line speaker bio. Next, they can select an upcoming event. After entering their details, they proceed to the 'SPECIAL' screen, where they can request food and accommodation, cab services, and specify their technical requirements. They then review and confirm their information before making a payment to complete the registration.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Contact - Email, LastName : Used to create a new contact record if the email doesn't exist.
- \* Speaker\_c - Email\_c, Event\_c, ... : Used to store speaker details and check for duplicate records.
- \* Event\_c - Upcoming\_Events\_c : Used to query upcoming events for speakers to choose from.







## **FLOW 4:**

### **ATTENDEE CANCELLATION FORM:**

**Process Type:** Screen Flow

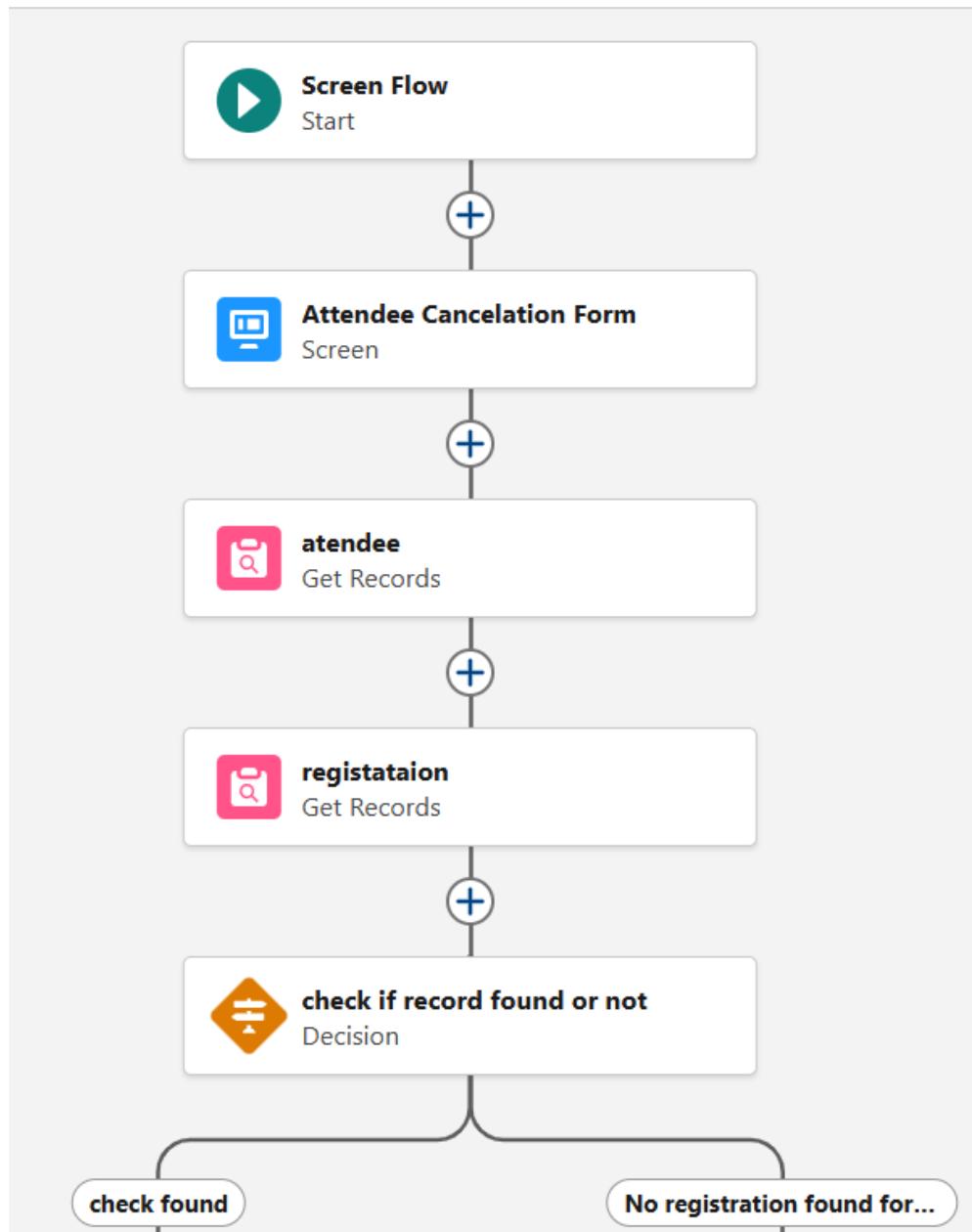
#### **DESCRIPTION:**

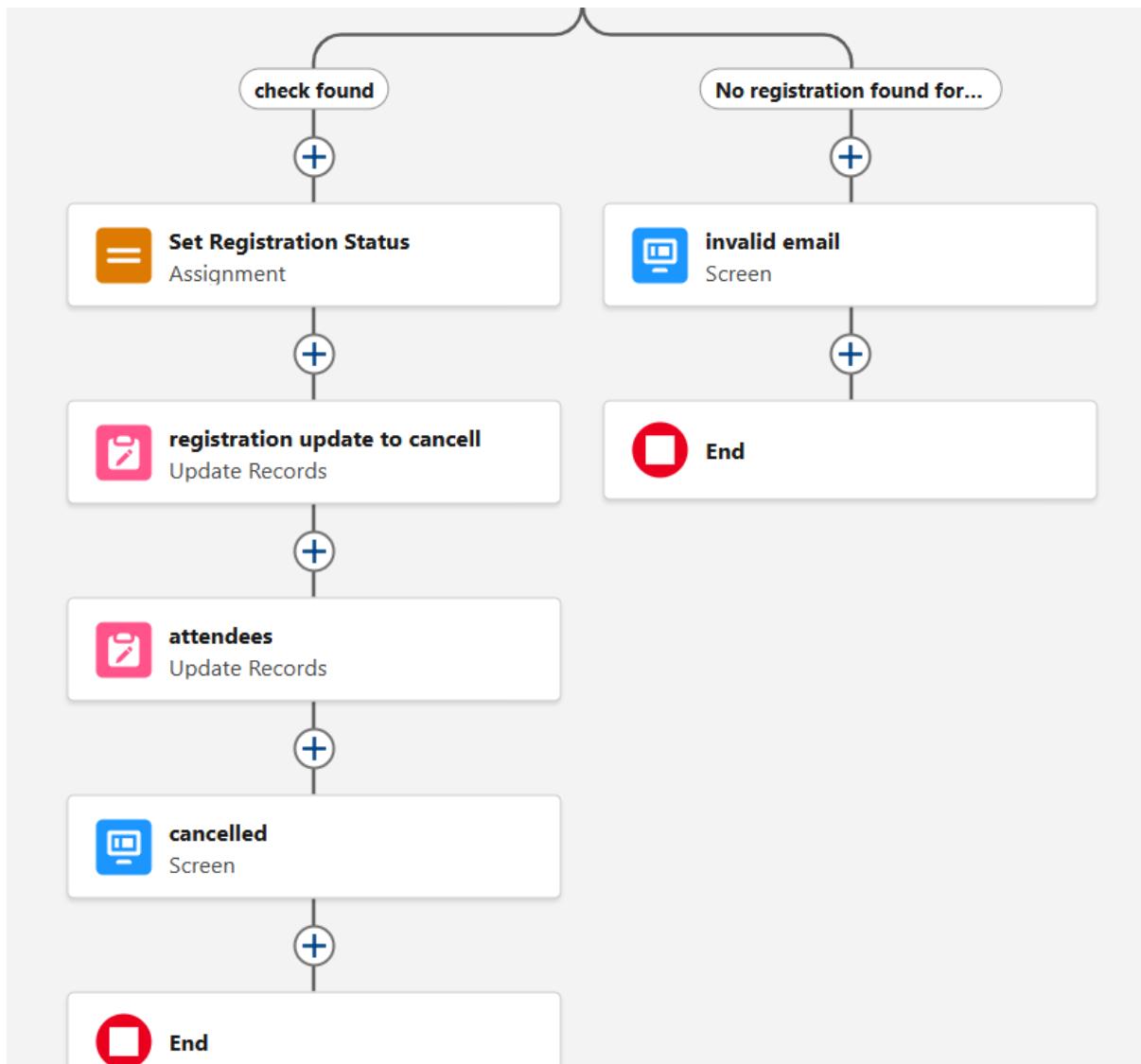
This is a Screen Flow.

This flow begins with a screen where attendees enter their registered email ID and phone number. It then checks if a matching record is found in the 'Attendee\_List\_c' object. If found, it proceeds to check if the attendee has a corresponding registration record in the 'Registration\_c' object. If both records are found, the flow updates the status of both records to 'Cancelled'. If no registration record is found, the flow displays an error message. If no attendee record is found, the flow ends without any action.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Attendee\_List\_c - Email\_c, Phone\_c, Status\_c : Used to lookup and update attendee records based on provided email ID and phone number.
- \* Registration\_c - Attendee\_c, Status\_c : Used to lookup and update registration records based on the associated attendee record.





## **FLOW 5:**

### **CANCELLED REGISTRATION COUNT:**

**PROCESS TYPE:** Autolaunched Flow

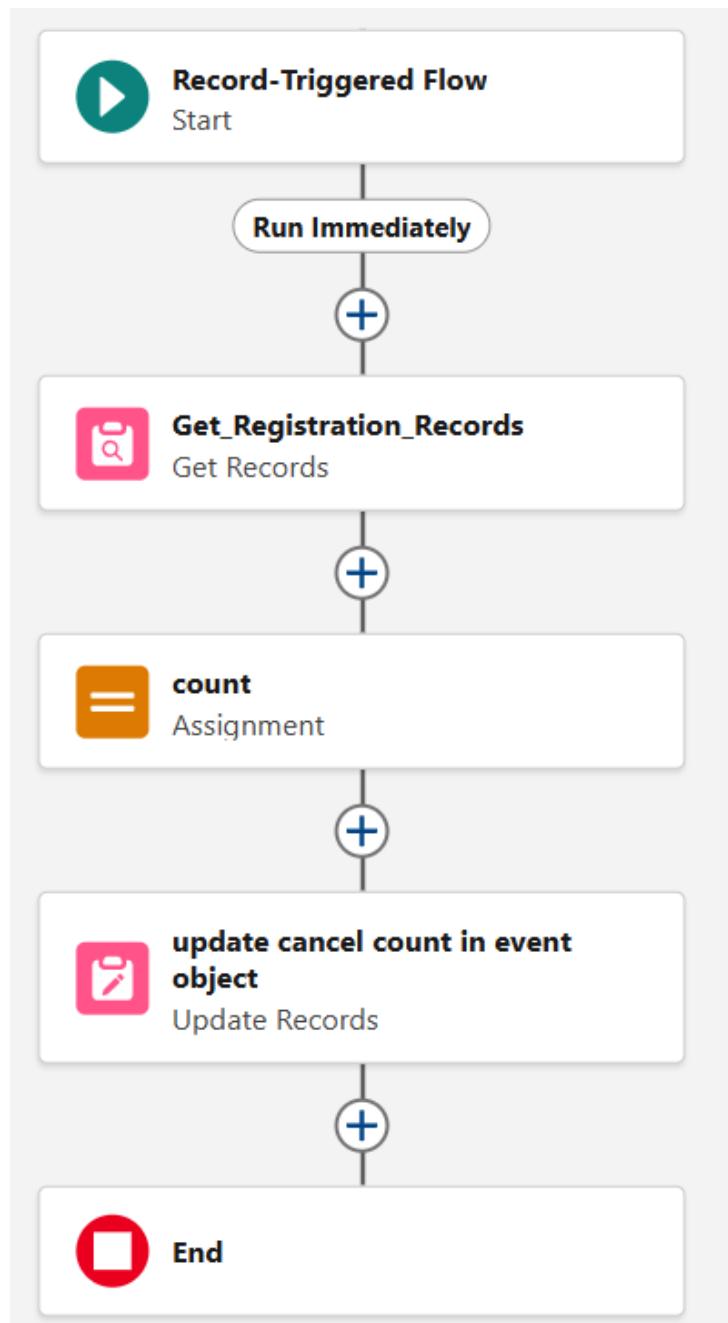
#### **DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after a Registration\_\_c record is updated. It checks if the status is 'Cancelled'. If so, it queries the Registration\_\_c records with the same event and cancelled status. Then, it updates the 'Cancelled\_Registration\_Count\_c\_\_c' field on the associated Event\_\_c record with the count of cancelled registrations.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Registration\_\_c - Status\_\_c : Checks for 'Cancelled' status
- \* Registration\_\_c - Event\_\_c : Uses to query related registrations
- \* Event\_\_c - Id : Uses to identify the event record
- \* Event\_\_c - Cancelled\_Registration\_Count\_c\_\_c : Updates with the count of cancelled registrations



## **FLOW 6:**

### **CANCELLATION EMAIL TO ATTENDEE:**

**PROCESS TYPE:** Autolaunched Flow

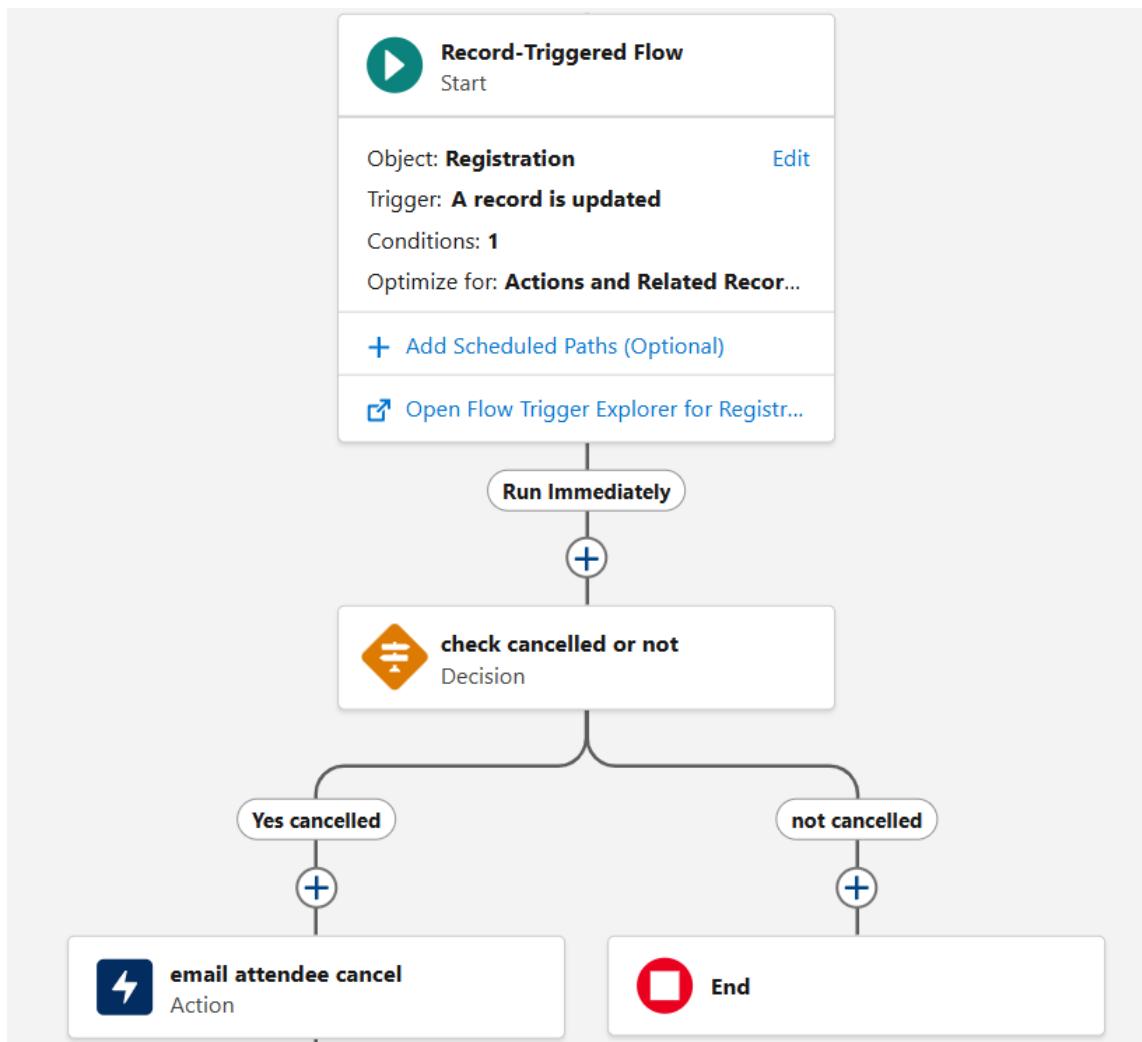
#### **DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after a record is updated on the 'Registration\_\_c' object. It checks if the status is 'Cancelled'. If yes, it sends an email to the attendee using the 'emailSimple' action, with the subject 'Registration Cancellation Confirmation' and a personalized body. The flow does not wait for the email action to complete.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

\* Registration\_\_c - Status\_\_c : The flow checks this field to determine if the registration status is 'Cancelled'.



## EMAIL FORMAT:

Hi {!\$Record.Attendee\_\_r.Name},

Your registration has been successfully cancelled.

If you have any questions, contact us at support@example.com.

Thank you.

## **FLOW 7:**

### **EVENT COLLECTION DATA TABLE:**

**PROCESS TYPE:** Screen Flow

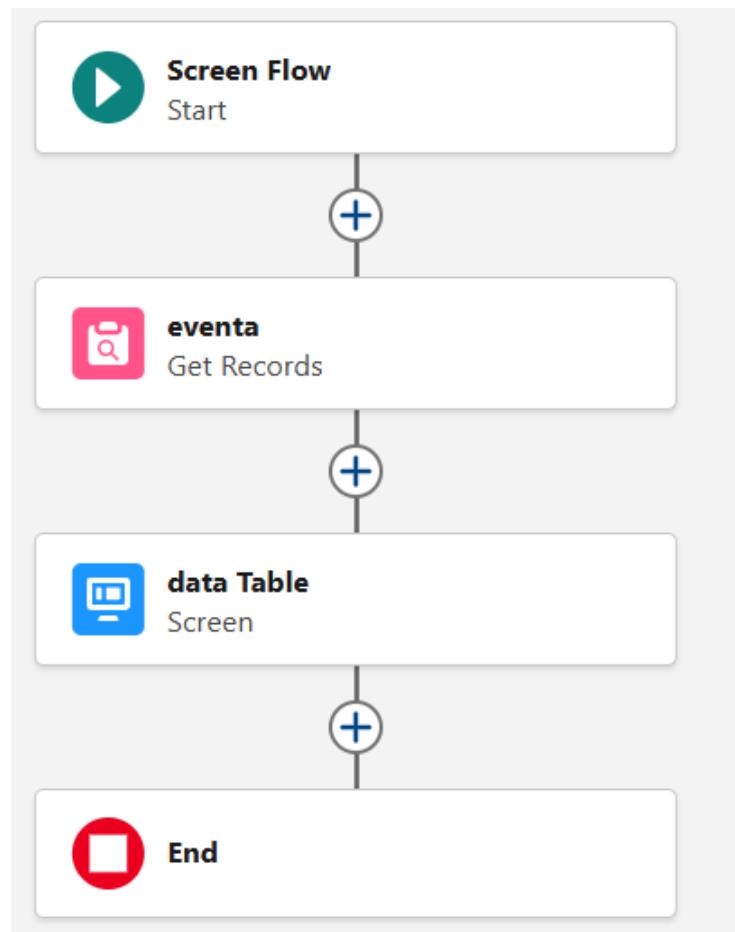
#### **DESCRIPTION:**

This is a Screen Flow.

This flow is triggered when a user navigates to a specific screen. It retrieves events from Salesforce and displays their details in a table on the screen. The flow does not require record changes to meet criteria and does not publish segments. It uses an 'Event\_\_c' object and filters out events where 'Name' or 'Date\_\_c' is null.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Event\_\_c - Name : Filters out events where 'Name' is null
- \* Event\_\_c - Date\_\_c : Filters out events where 'Date\_\_c' is null



## **FLOW 8:**

### **PAYMENT DETAILS FOR ATTENDEE AFTER REGISTRATION:**

**PROCESS TYPE:** Autolaunched Flow

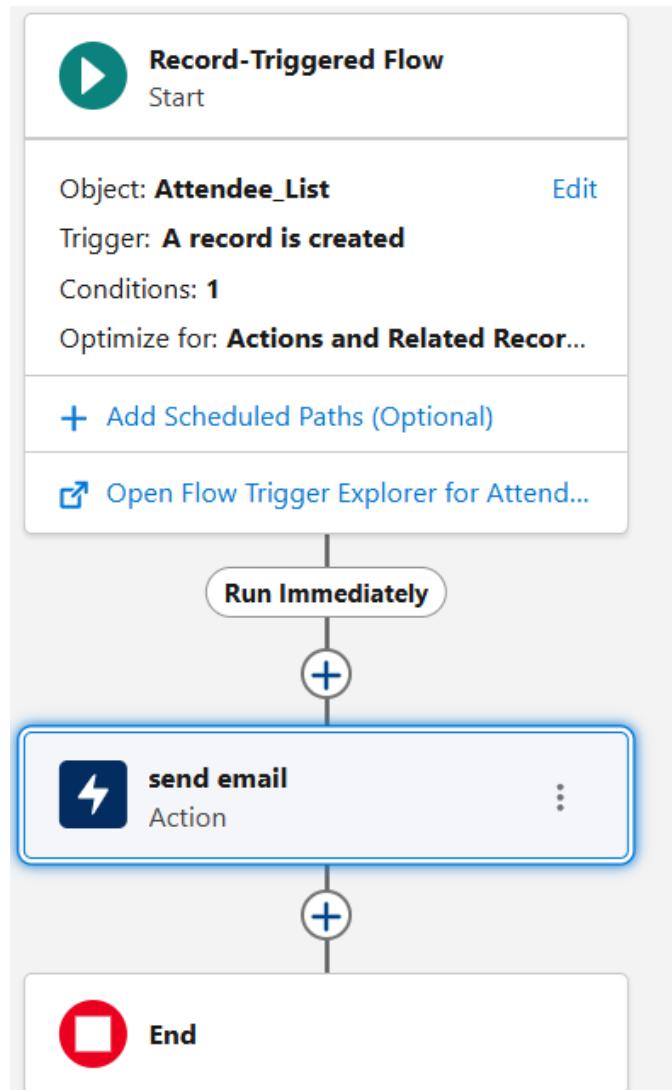
#### **DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after a record is created on the 'Attendee\_List\_\_c' object. It checks if the 'Attende\_Payment\_Confirmed\_\_c' field is false. If true, it sends an email to the attendee using their email address stored in the 'Email\_\_c' field. The email contains details about the total amount to be paid, payment breakdown, and bank details for payment. It also includes a note to share the transaction reference with the support team.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Attendee\_List\_\_c - Attende\_Payment\_Confirmed\_\_c : The flow checks this field to determine if an email needs to be sent.
- \* Attendee\_List\_\_c - Email\_\_c : The flow uses this field to populate the recipient's email address in the email action.



### EMAIL FORMAT:

Dear {!\$Record.Name},

Thank you for registering for the event!

We've received your registration details.

Your spot is currently reserved, and will be confirmed once your payment is verified by our finance team.

Kindly pay your total amount of ₹{!\$Record.Final\_Amount\_\_c} to confirm your registration.

### Payment Breakdown:

**Basic Event Fee :** {!\$Record.Basic\_Fee\_payment\_\_c}

**Final Event Fee :** {!\$Record.Final\_Fee\_Payment\_c}

**cabFee:** {!\$Record.cab\_Fee\_c}

**Food Bill :** {!\$Record.FoodBill\_c}

## **FLOW 9:**

### **EVENT REGISTRATION PAYMENT VERIFICATION:**

**PROCESS TYPE:** Autolaunched Flow

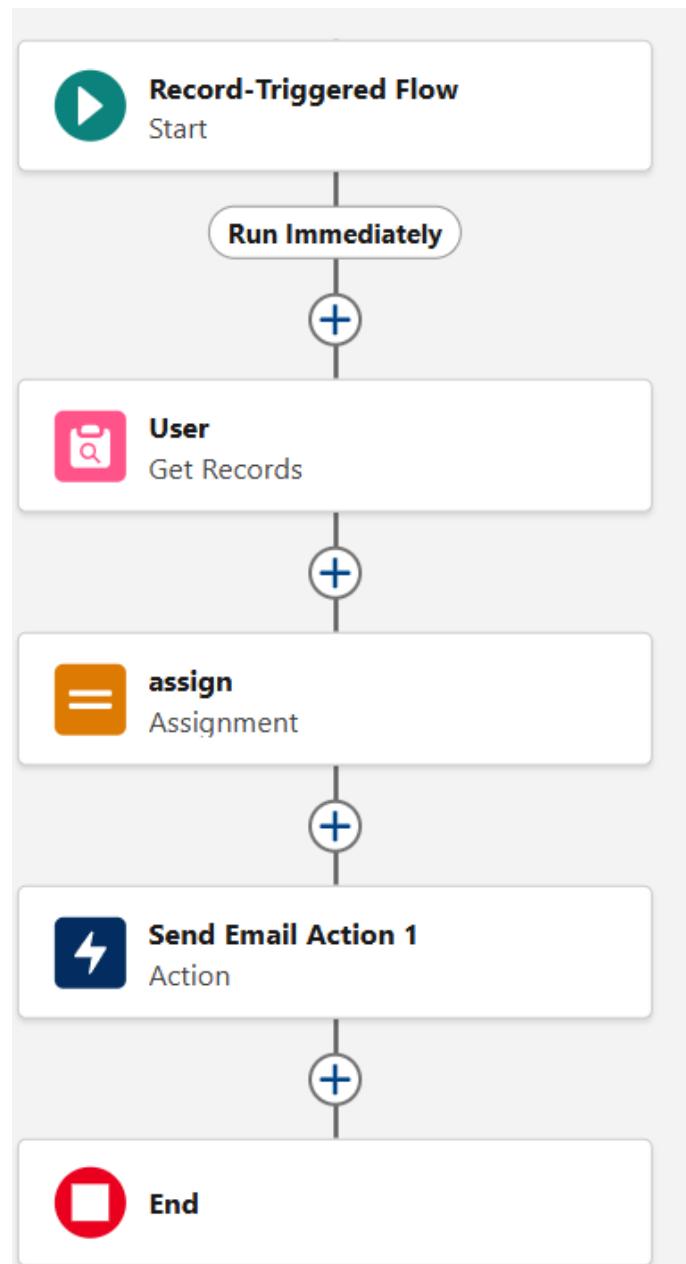
#### **DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after a record is created on the 'Attendee\_List\_\_c' object. It checks if the 'Attendee\_Payment\_Confirmed\_\_c' field is false. If true, it looks up the User record with the profile ID '00egK000003r0WP' and assigns the user's email to a variable. Then, it sends an email to the finance team using the 'emailSimple' action, with the subject 'Payment Confirmation New Attendee is Created' and a custom HTML body containing attendee details.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Attendee\_List\_\_c - Attendee\_Payment\_Confirmed\_\_c : Checks if this field is false to trigger the flow.
- \* User - Email : Uses this field to send an email to the finance team.



**EMAIL FORMATE:**

Dear Financial Team,

⚠ A new attendee has registered for the event. Kindly verify and update the payment status in the system.

- ◆ Attendee Name : {!\$Record.Name}
- ◆ Registration ID: {!\$Record.Id}

Please take the necessary action at your earliest convenience.

Thank you,

Event Registration System

## **FLOW 10:**

### **ATTENDEE PAYMENT CONFIRMATION NOTIFICATION:**

**PROCESS TYPE:** Autolaunched Flow

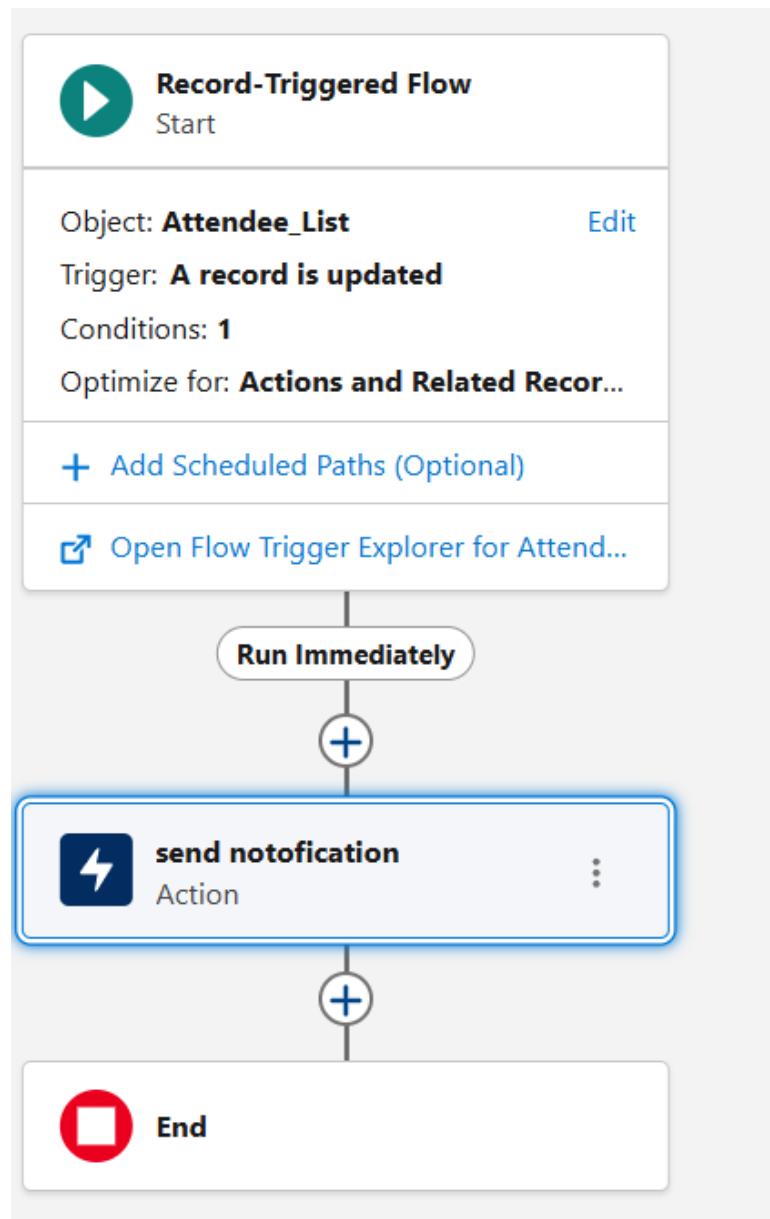
#### **DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after an update on the Attendee\_List\_\_c object. It checks if the Attendee\_Payment\_Confirmed\_\_c field is true. If so, it sends a Chatter post with the record ID and attendee name.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Attendee\_List\_\_c - Attendee\_Payment\_Confirmed\_\_c : The flow checks this field to determine if a Chatter post should be sent.
- \* User - Id : The flow uses the user's ID to send the Chatter post.



Post to Chatter Formate:

- Attendees payment has been confirmed. Record ID: {!\$Record.Id} Name of Attendee :{!\$Record.Name}

## **FLOW 11:**

### **CONFIRMATION TO ATTENDEE AFTER PAYMENT:**

**PROCESS TYPE:** Autolaunched Flow

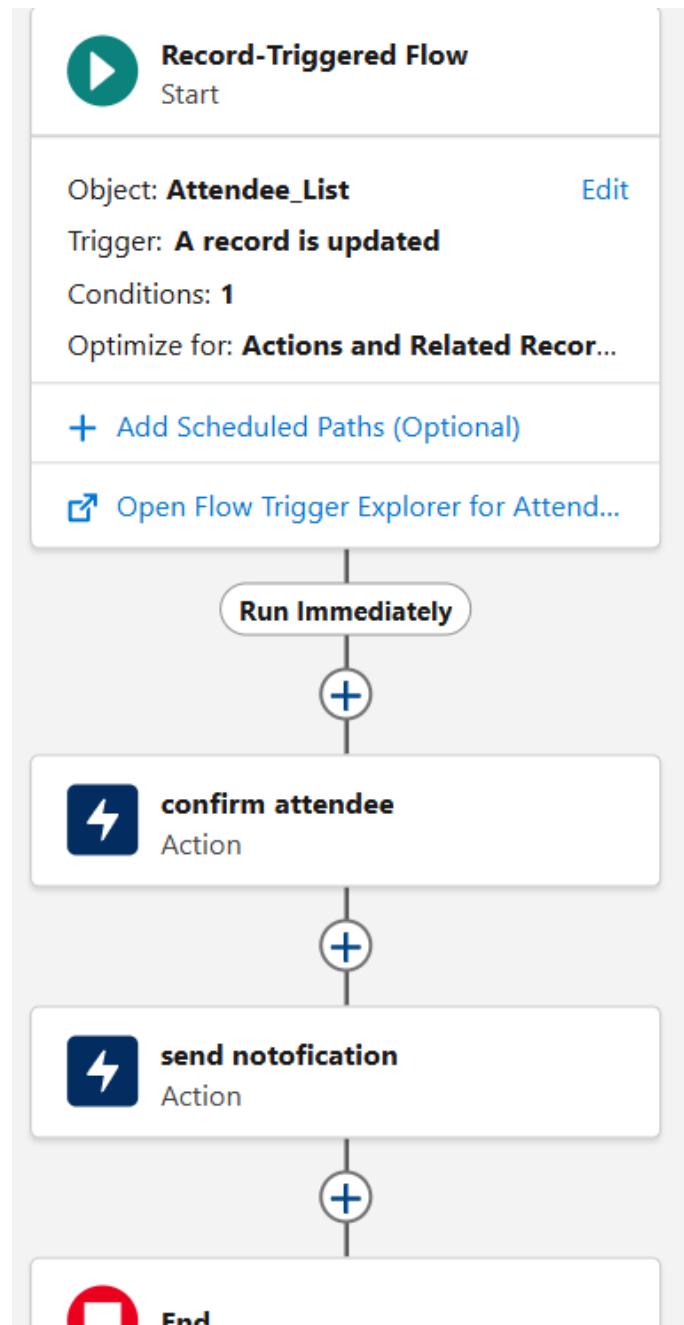
#### **DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after an update on the Attendee\_List\_\_c record. It checks if the Attendee\_Payment\_Confirmed\_\_c field is true. If yes, it performs two actions: sends a confirmation email to the attendee using their email address stored in the Email\_\_c field, and posts a message on the specified Chatter group with the attendee's name and record ID.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Attendee\_List\_\_c - Attendee\_Payment\_Confirmed\_\_c : The flow is triggered by updates to this field.
- \* Attendee\_List\_\_c - Email\_\_c : The email address from this field is used to send the confirmation email.
- \* Chatter Group - Name : The flow posts a message to this group.



## **EMAIL FORMATE:**

Hi {\$Record.Name},  

Great news! Your payment has been successfully confirmed, and your spot for the event is officially reserved.



Event Details:

Event: {\$Record.Events\_\_r.Name}

Date: {\$Record.Events\_\_r.Date\_\_c}



What's Next?

- Bring your confirmation email or ID on the event day.
- Arrive 15 minutes early for check-in.
- Get ready for an exciting experience!

If you have any questions, feel free to reply to this email or contact our support team.

Thanks for being part of this event!

Warm regards,

Event Coordination Team

## **NOTIFICATION FORMAT:**

 Attendees payment has been confirmed. Record ID: {\$Record.Id} Name of Attendee :{\$Record.Name}

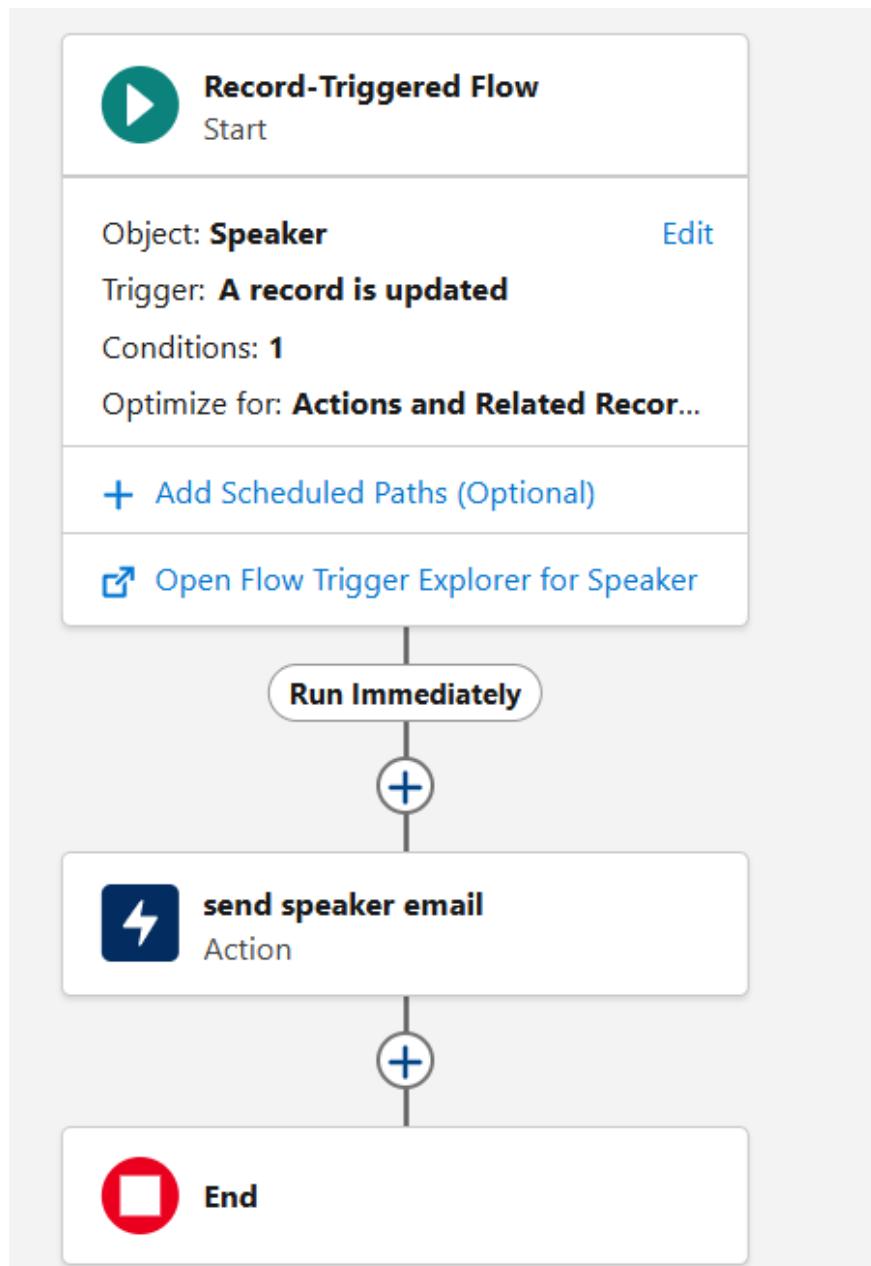
**FLOW 12:****CONFORMATION MAIL TO SPEAKER:****PROCESS TYPE :Autolaunched Flow****DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after a record is updated on the 'Speaker\_\_c' object. It checks if the 'Approved\_Speaker\_\_c' field is true. If so, it sends an email to the speaker using the 'Email\_\_c' field from the record. The email includes the event name, date, and a personalized message welcoming the speaker.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Speaker\_\_c - Approved\_Speaker\_\_c : The flow checks this field to determine if the speaker is approved.
- \* Speaker\_\_c - Email\_\_c : The flow uses this field to send the email to the speaker.



## **EMAIL FORMATE:**

Hello {!\$Record.Name}, 

 Great news! Your registration has been approved by the event manager, and you're now officially part of our speaker lineup!, and you're now officially part of our speaker lineup! 

Event Name: {!\$Record.Event\_r.Name}

Date Of Event : {!\$Record.Event\_r.Date\_c}

 Thank you for joining us!

We're thrilled to have you on board and can't wait for the energy and insights you'll bring to your session! 

If you have any questions or need support, feel free to reply to this email — we're here to help. 

Warm regards,

Event Team

If you have any questions or need support, feel free to reply to this email.

{!\$Record.Event\_r.Name}

 [SupportEventsSpot@gmail.com](mailto:SupportEventsSpot@gmail.com)

## **FLOW 13:**

### **Check ATTENDEE SEAT AVAILABILITY:**

**PROCESS TYPE:** Screen Flow

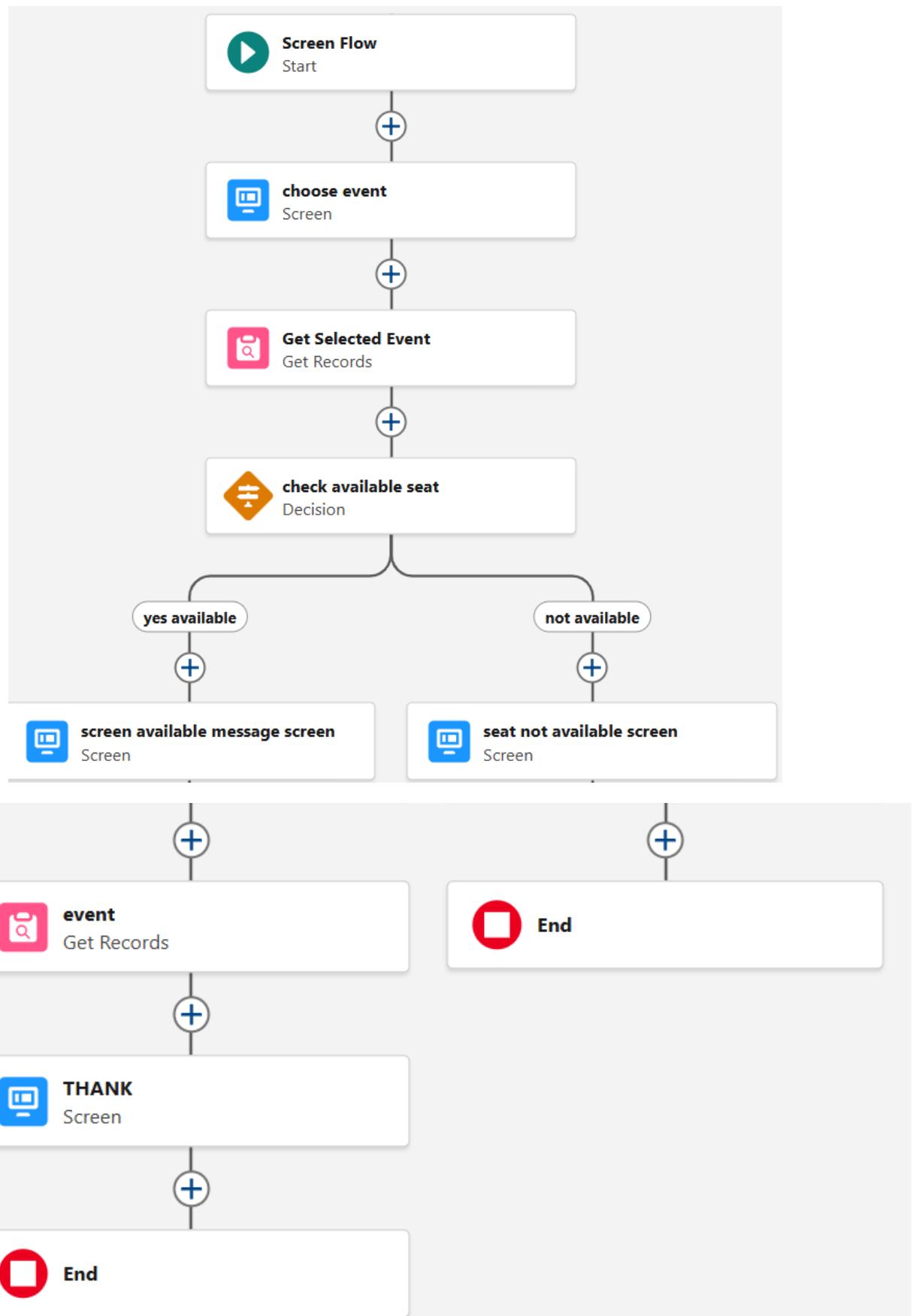
#### **DESCRIPTION:**

This is a Screen Flow.

The flow begins with a screen where the user selects an event from a dropdown list. After the user makes a selection, the flow checks if seats are available for the chosen event. If seats are available, it displays a message screen with the available seats. If not, it shows a screen indicating that seats are not available. Finally, it displays a thank you screen.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

\* Event\_c - Id, Name, Available\_Seats\_c : The flow queries and uses these fields to retrieve and display event information.



**FLOW 14:****SPEAKER SEAT AVAILABILITY:**

**PROCESS TYPE:** Screen Flow

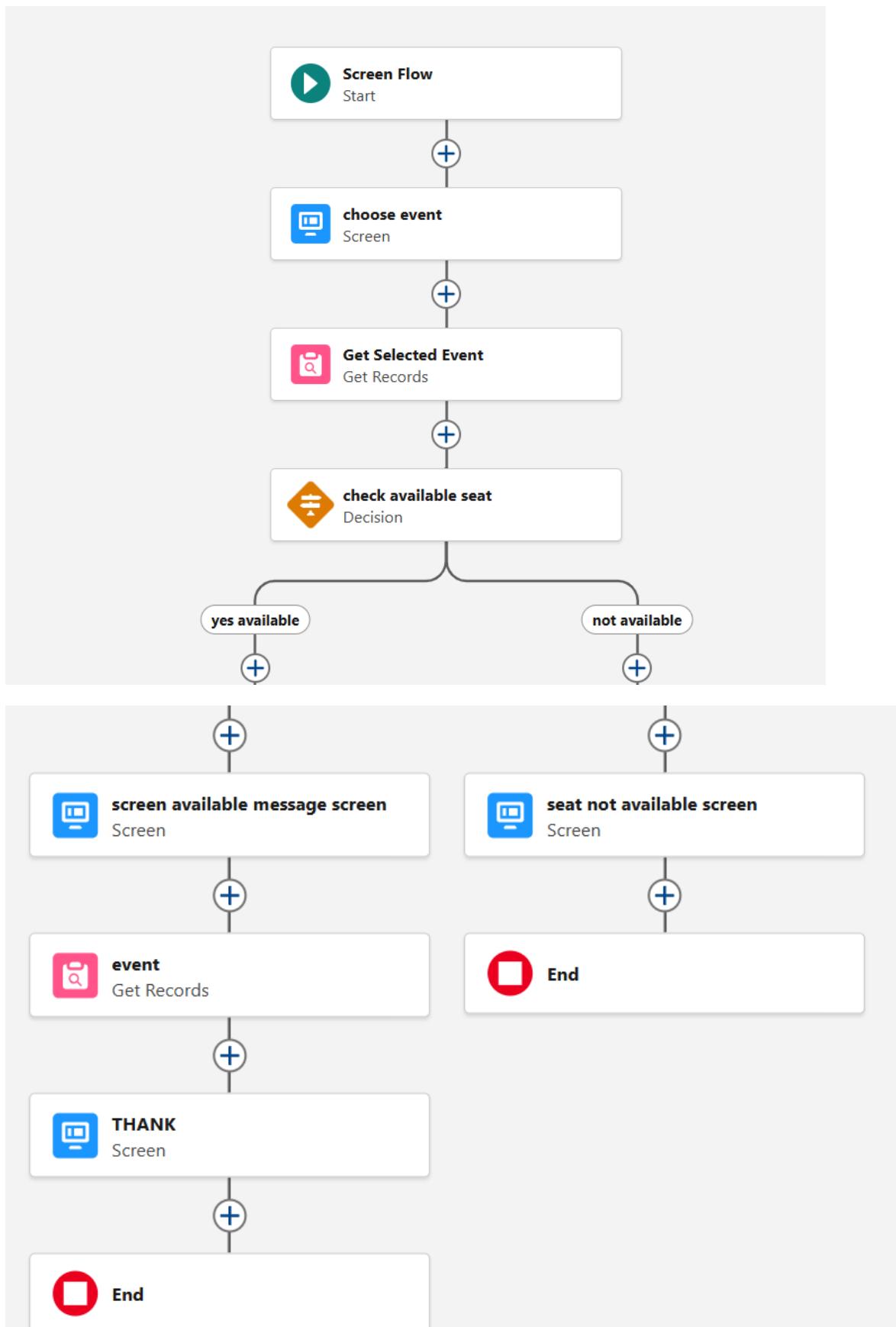
**DESCRIPTION:**

This is a Screen Flow.

The flow begins with a screen where the user selects an event from a dropdown list. Once the event is chosen, the flow checks if seats are available for that event. If seats are available, it displays a message screen showing the number of available seats. If seats are not available, it displays a different message screen indicating the same. After displaying the message, the flow ends with a thank you screen.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

\* Event\_c - Id, Name, available\_set\_for\_speakers\_c : The flow queries and uses these fields from the Event\_c object to retrieve the selected event and check the availability of seats.



## **FLOW 15:**

### **REFUND SPEAKER AMOUNT FLOW:**

**PROCESS TYPE:** Autolaunched Flow

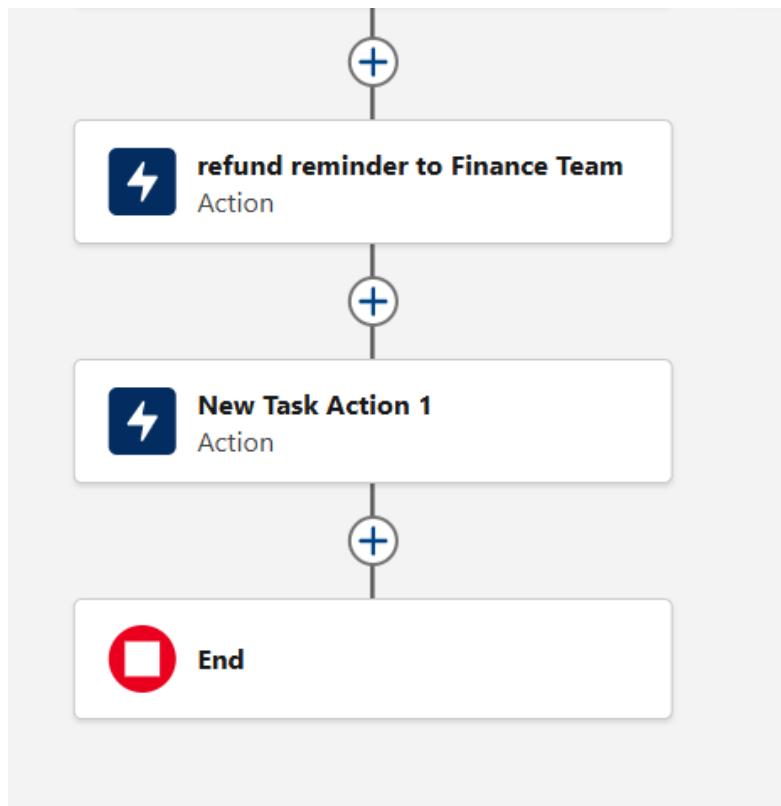
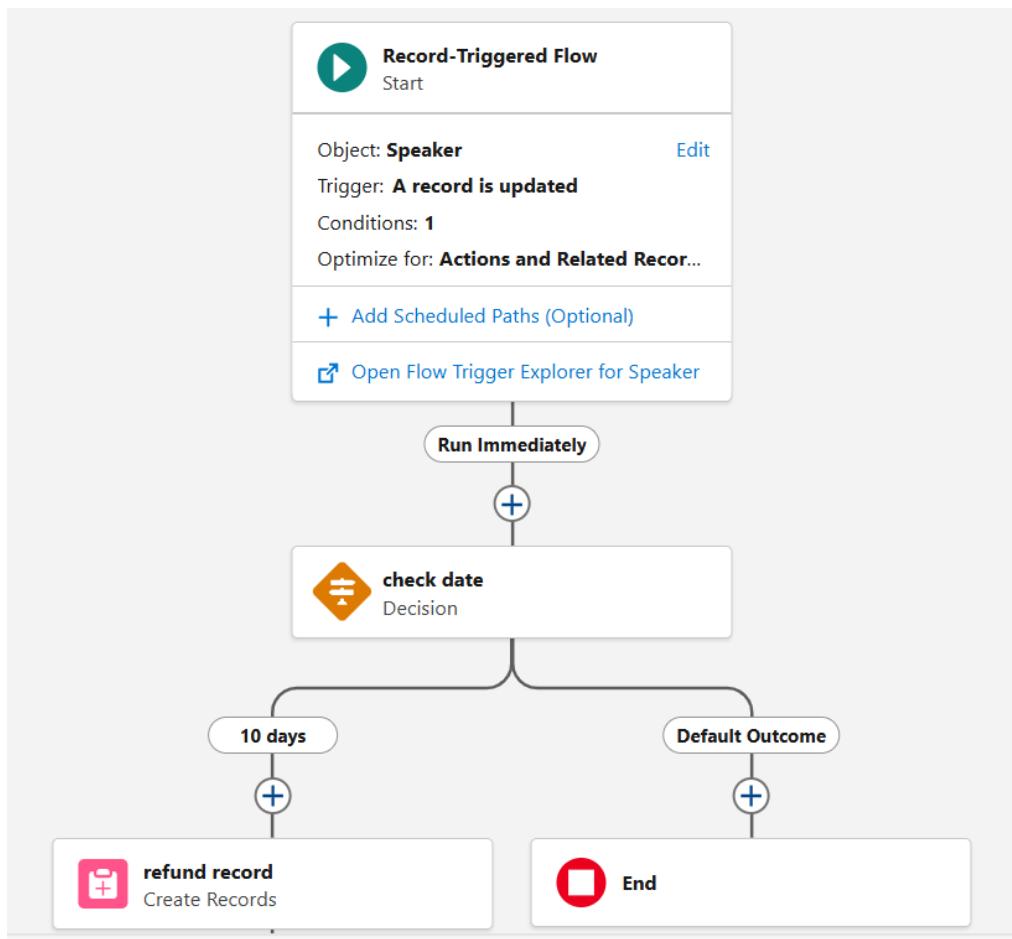
#### **DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after a record is updated on the 'Speaker\_\_c' object. It checks if the 'Need\_To\_Refund\_\_c' field is true. If so, it calculates the number of days left until the event and compares it to 10. If the event is 10 days or less away, it creates a new 'Refund\_Request\_\_c' record and posts a Chatter message to the Finance Team, along with creating a task for them.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Speaker\_\_c - Need\_To\_Refund\_\_c : Checks if this field is true to start the flow.
- \* Speaker\_\_c - Speaker\_Payment\_Confirm\_\_c : Uses this field to ensure payment confirmation.
- \* Refund\_Request\_\_c - Amount\_\_c : Populates this field with the total amount from the Speaker record.



**POST TO CHATTER FORMATE:**

Speaker {!\$Record.Name} Record ID: {!\$Record.Id} , was not approved for participation. Amount Paid: ₹ {!\$Record.Total\_Amount\_\_c} Please initiate the refund process.

## **FLOW 16:**

### **SPEAKER REFUND SUCCESS EMAIL :**

**Process Type:** Autolaunched Flow

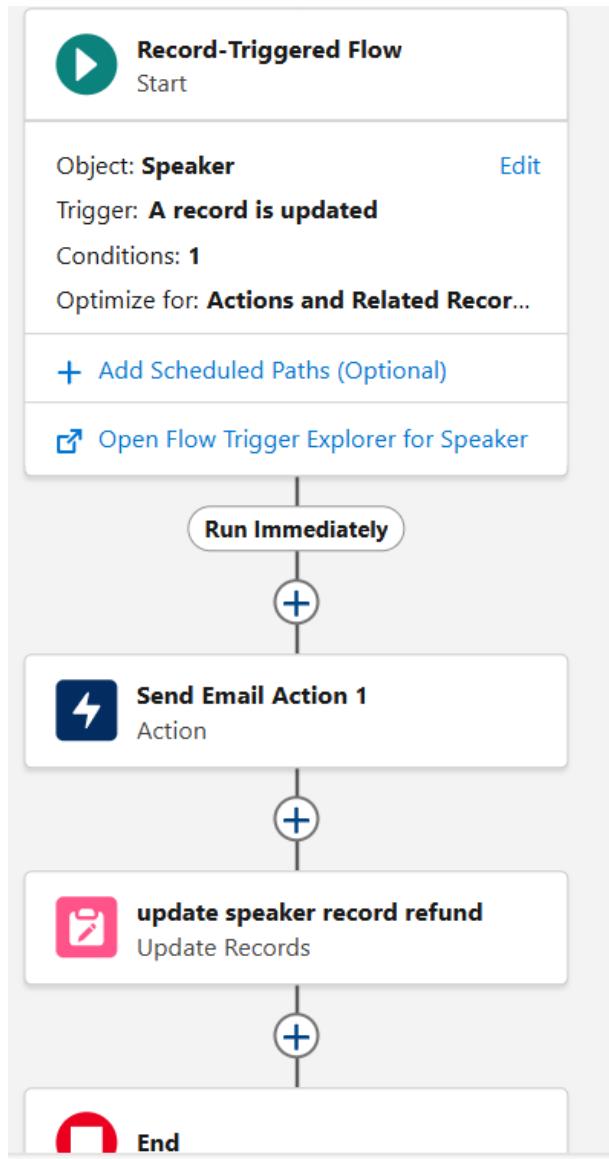
#### **DESCRIPTION:**

This is a Record-Triggered Flow.

This flow is triggered after a record is saved on the Speaker\_\_c object. It checks if the Refunded\_\_c field is true. If it is, the flow sends an email using the emailSimple action, populating the recipient's email address, subject, and body with relevant information. After sending the email, the flow updates the Refund\_Request\_\_c record by setting the Refunded\_\_c field to true.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Speaker\_\_c - Refunded\_\_c : The flow checks this field to determine if an email should be sent.
- \* Refund\_Request\_\_c - Refunded\_\_c : The flow updates this field to indicate that the refund process is complete.



## EMAIL FORMAT:

Dear {\$Record.Name},

your refund of ₹ {\$Record.Total\_Amount\_c} for the event {\$Record.Event\_r.Name} has been successfully processed.

✉ Reason: **Speaker not approved for participation**

✉ Refund Status: ✓ **Completed**

Thank you for your patience and understanding. 👉

If you have any questions, feel free to reach out to our team.

Warm regards,

 Finance Team

 [evenspotFinance@gmail.com](mailto:evenspotFinance@gmail.com)

## FLOW 17:

### EVENT REGISTERED FLOW TO SPEAKER:

**PROCESS TYPE:** Autolaunched Flow

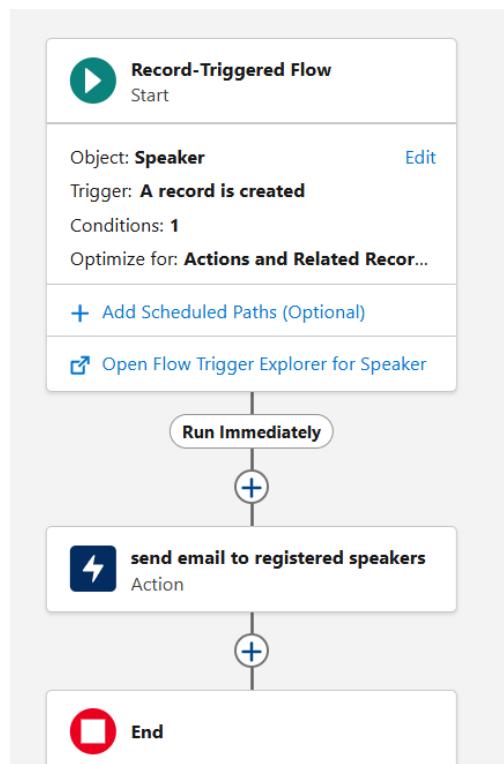
#### DESCRIPTION:

This is a Record-Triggered Flow.

This flow is triggered when a new 'Speaker\_\_c' record is created and the 'Speaker\_Payment\_Confirm\_\_c' field is true. It then sends an email to the registered speaker using the 'Email\_\_c' field from the record. The email includes a personalized message, details about the next steps, what they should know, and contact information for support.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Speaker\_\_c - Email\_\_c : Used to send the email to the registered speaker.
- \* Speaker\_\_c - Speaker\_Payment\_Confirm\_\_c : Used to filter records where payment confirmation is true.



## **EMAIL FORMATE:**

Hi {\$Record.Name}, 

Thank you for registering as a speaker for {\$Record.Event\_r.Name}! 

We're excited to have you on board.

### **What Happens Next**

-  Your registration status is now "Registered".
-  A confirmation of your payment has been received.
-  Your profile will now be reviewed by our Event Manager.

### **What You Should Know**

-  Approval is not guaranteed.

The manager will assess your topic, profile, and relevance to the event.

-  If approved, you'll receive a final confirmation and be added to the speaker list.
-  If not approved, your payment will be refunded to your original method 10 days before the event date.

### **Need Help?**

If you have any questions or need assistance, feel free to reply to this email or contact our team.

Thank you once again for your interest and enthusiasm!

 Contact Number: +91 98765 43210

 Email: [support@eventspot.com](mailto:support@eventspot.com)

We look forward to reviewing your submission. 

Warm regards,

Event Coordination Team

## **FLOW 18:**

## SPEAKER CERTIFICATE GENERATION FLOW:

**Process Type:** Autolaunched Flow

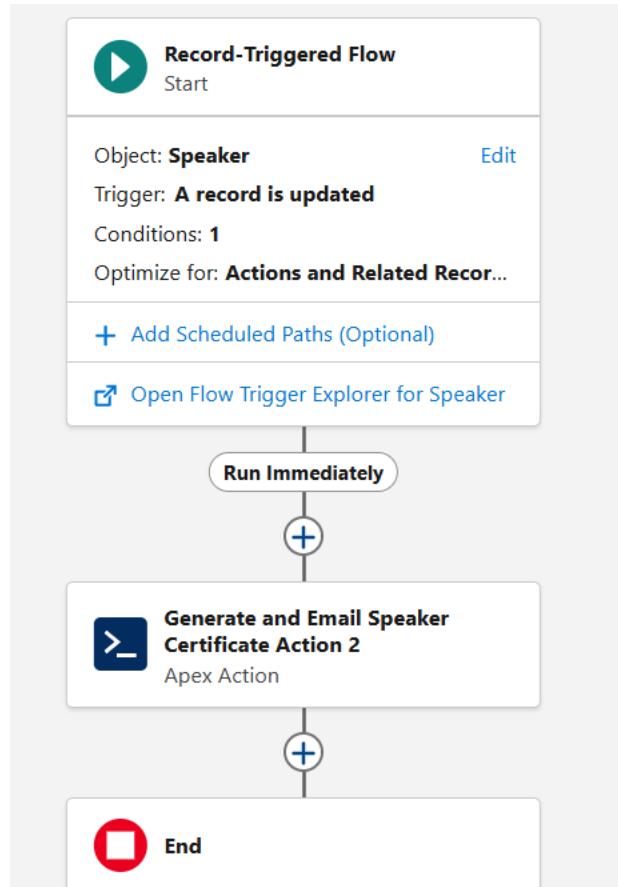
### DESCRIPTION:

This is a Record-Triggered Flow.

This flow is triggered after an update on the Speaker\_\_c object. It checks if the Attended\_\_c field is true. If so, it calls an Apex action named 'CertificateGenerator' with the current record's ID as input. This flow does not wait for the Apex action to complete and does not store its output automatically.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

\* Speaker\_\_c - Attended\_\_c : The flow checks this field to determine if the certificate generation action should be performed.



**FLOW 19:**

## **FEEDBACK FORM ATTENDEE:**

**PROCESS TYPE:** Screen Flow

### **DESCRIPTION:**

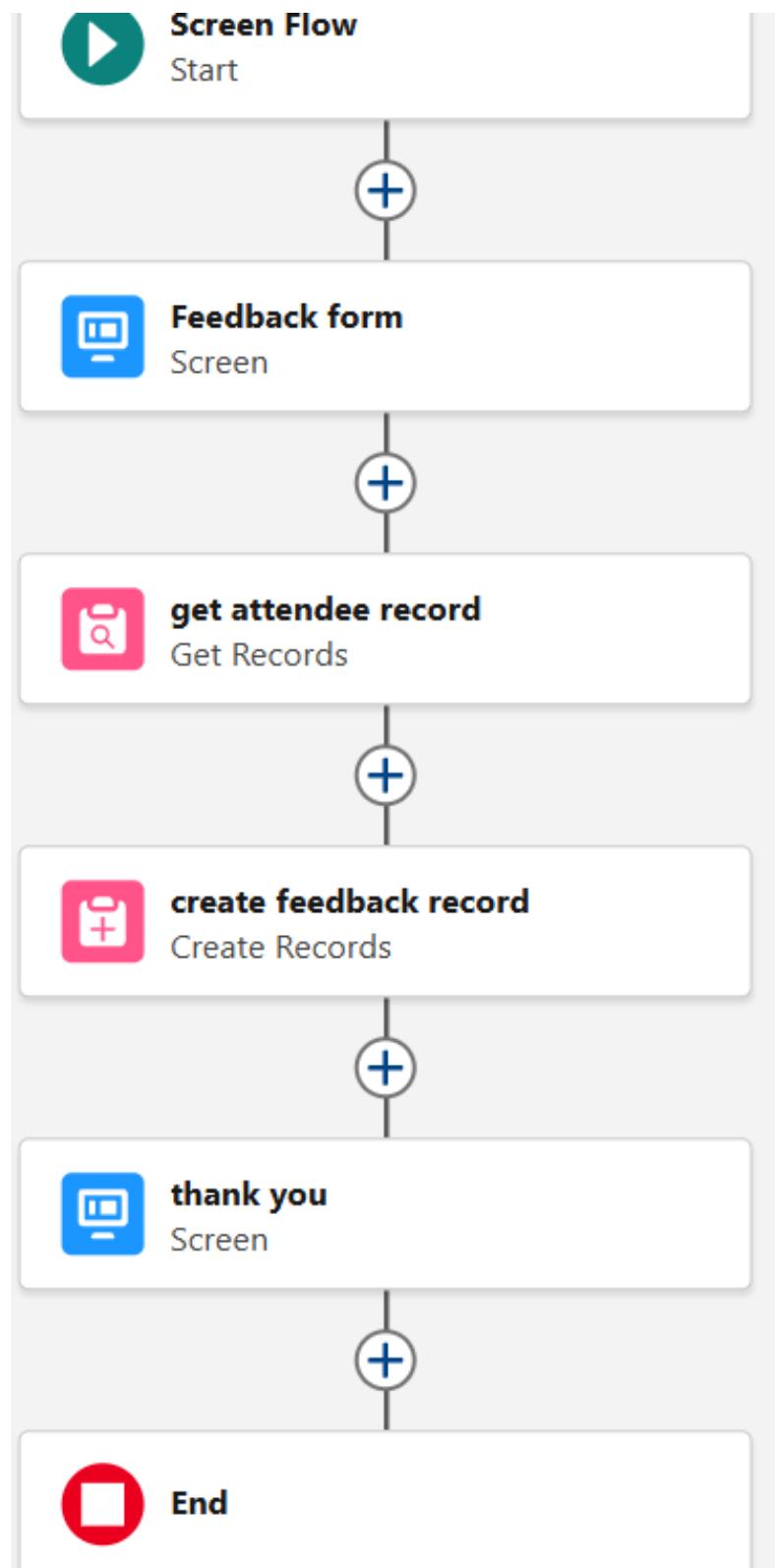
This is a Screen Flow.

This flow is triggered when a user navigates to a specific screen. It presents a form where attendees can enter their name, choose an event, rate the speaker and event organization, share their overall experience, and provide additional comments. The flow then creates a new feedback record using the provided information and displays a thank you message.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

\* Attendee\_List\_c - Name : Used to query attendee record based on entered name.

\* Feedback\_c - ['Attendee\_List\_c', 'Comments\_c',  
'Event\_Organization\_Rating\_c', 'Event\_c', 'Overall\_Experience\_c',  
'Speaker\_Rating\_c'] : Used to create a new feedback record with the collected data.



## FLOW 20:

### FEEDBACK TO ATTENDEE EMAIL:

**PROCESS TYPE:** AutoLaunched Flow

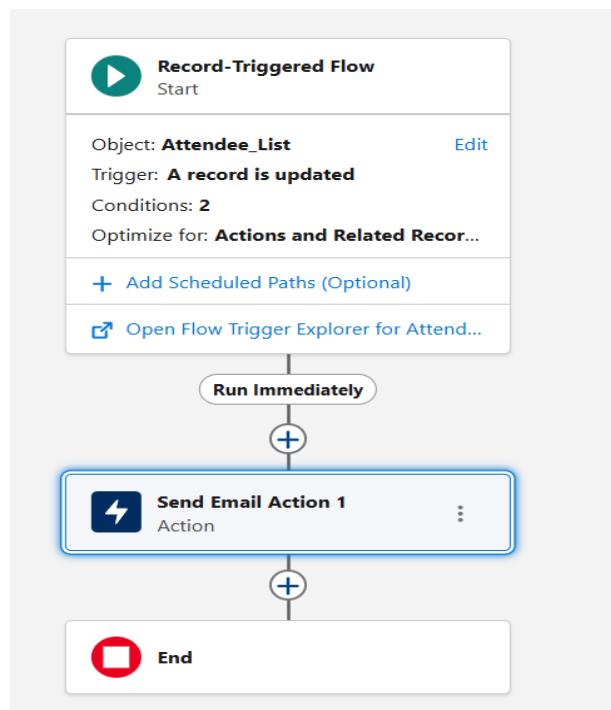
#### DESCRIPTION:

This is a Record-Triggered Flow.

This flow triggers when an Attendee\_List\_\_c record is updated. It checks if the attendee has attended the event and their status is 'Attended'. If true, it sends an email using the attendee's email address, with a predefined subject and body containing a feedback form link.

Several objects and their fields worth mentioning. Below are the details (the format is given as: object\_name - field\_name: interaction)

- \* Attendee\_List\_\_c - Attended\_\_c : Checks if the attendee has attended the event
- \* Attendee\_List\_\_c - Status\_\_c : Verifies the attendee's status is 'Attended'
- \* Attendee\_List\_\_c - Email\_\_c : Uses the attendee's email address to send the email



**EMAIL FORMATE:**

Hi {\$Record.Name},

Thank you for attending {\$Record.Events\_\_r.Name}!

We hope you had a great experience. Your feedback is very important to us and helps us improve future events.

👉 Please take a moment to fill out our feedback form:

<https://orgfarm-07fc65cb7b-dev-ed.develop.my.site.com/EventForm/>

It only takes 2 minutes, and your input means a lot!

We'd love to hear your thoughts on the event, the speakers, and the overall experience. Your feedback helps us make future events even better.

Thank you again for being part of our event.

Warm regards,

Team EventSpot Pvt Ltd