Phase 4 Report: Process Automation (Admin)

Project: Hyper-Local 'Barter & Skill-Share' Network Management

Batch: [Your Batch Number]

Program: TCS Last Mile SmartBridge

Prepared by: [Your Name]

1. Introduction

This phase focuses on implementing business process automation using Salesforce's declarative tools. Automation is crucial for improving efficiency, reducing manual data entry, and ensuring data consistency. For this project, we utilized Salesforce Flow to automate a key process within the NeighborNet application, enhancing the user experience for the system administrators.

2. Objectives

- To identify a key business process suitable for automation.
- To design and build a Record-Triggered Flow to automate the process.
- To ensure the automation runs only when specific criteria are met.
- To test the automation thoroughly to verify its functionality and impact on related records.

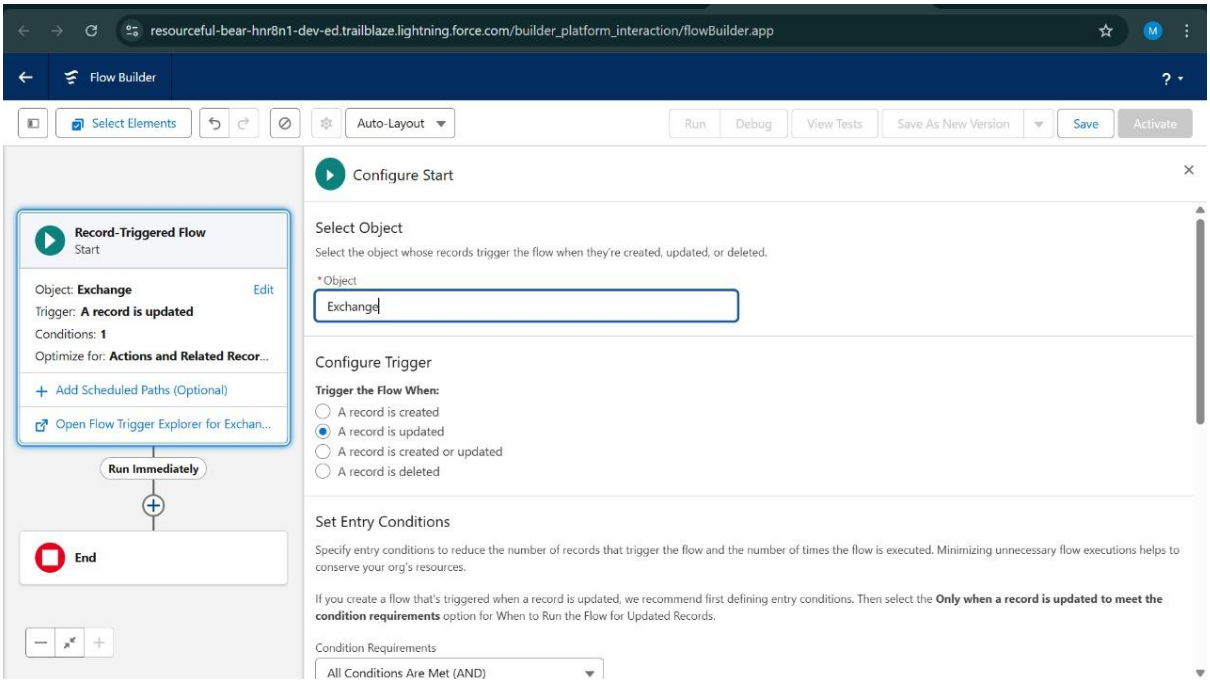
3. Automation Scenario: Update Offer Status on Exchange Completion

Description: In our application, when an Exchange between two members is successfully completed, the parent Offer should reflect this change. Manually updating the Offer status to "Fulfilled" is an extra step for the admin that can be forgotten, leading to data inconsistency.
Solution: We created a Record-Triggered Flow that automatically listens for an Exchange record's Status field to be updated to "Completed". When this happens, the Flow finds the related parent Offer record and updates its Status field to "Fulfilled".

4. Steps Performed

4.1 Flow Trigger Configuration

A Record-Triggered Flow was configured to run when an Exchange__c record is updated. Entry criteria were set to ensure the Flow only triggers when the Status__c field is updated to "Completed," optimizing performance and ensuring the automation runs only when intended.


resourceful-bear-hnr8n1-dev-ed.trailblaze.lightning.force.com/builder_platform_interaction/flowBuilder.app
Flow Builder
Select Elements
Auto-Layout
Run
Debug
View Tests
Save As New Version
Save
Activate
Edit
Run Immediately

Record-Triggered Flow
Start

Object: **Exchange**
Trigger: **A record is updated**
Conditions: **1**
Optimize for: **Actions and Related Recor...**

+ Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Exchan...

End

Configure Start

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

*Object
Exchange

Configure Trigger

Trigger the Flow When:

○ A record is created
● A record is updated
○ A record is created or updated
○ A record is deleted

Set Entry Conditions

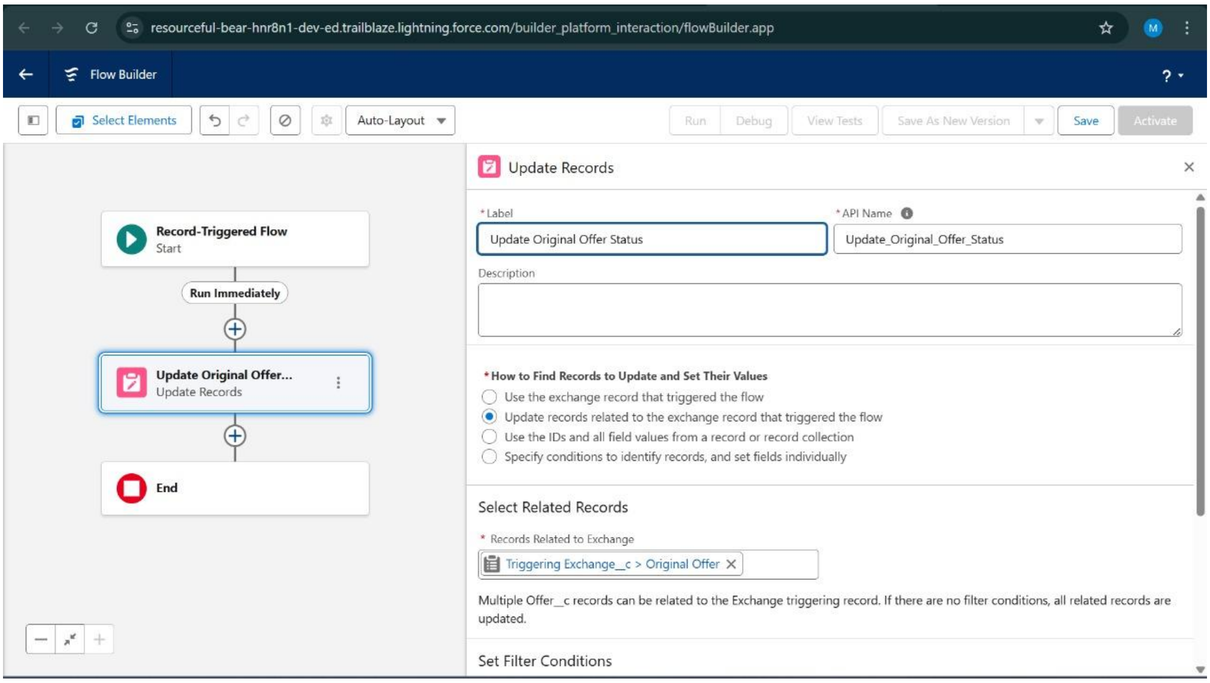
Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements
All Conditions Are Met (AND)

4.2 Update Related Record Element

An "Update Related Records" element was added to the Flow. This element was configured to find the parent Offer__c record via the Original_Offer__r relationship and set its Status__c field to the value "Fulfilled". This declarative step efficiently handles the cross-object update without any code.



Update Records

*Label: Update Original Offer Status

*API Name: Update_Original_Offer_Status

Description

*How to Find Records to Update and Set Their Values

○ Use the exchange record that triggered the flow
● Update records related to the exchange record that triggered the flow
○ Use the IDs and all field values from a record or record collection
○ Specify conditions to identify records, and set fields individually

Select Related Records

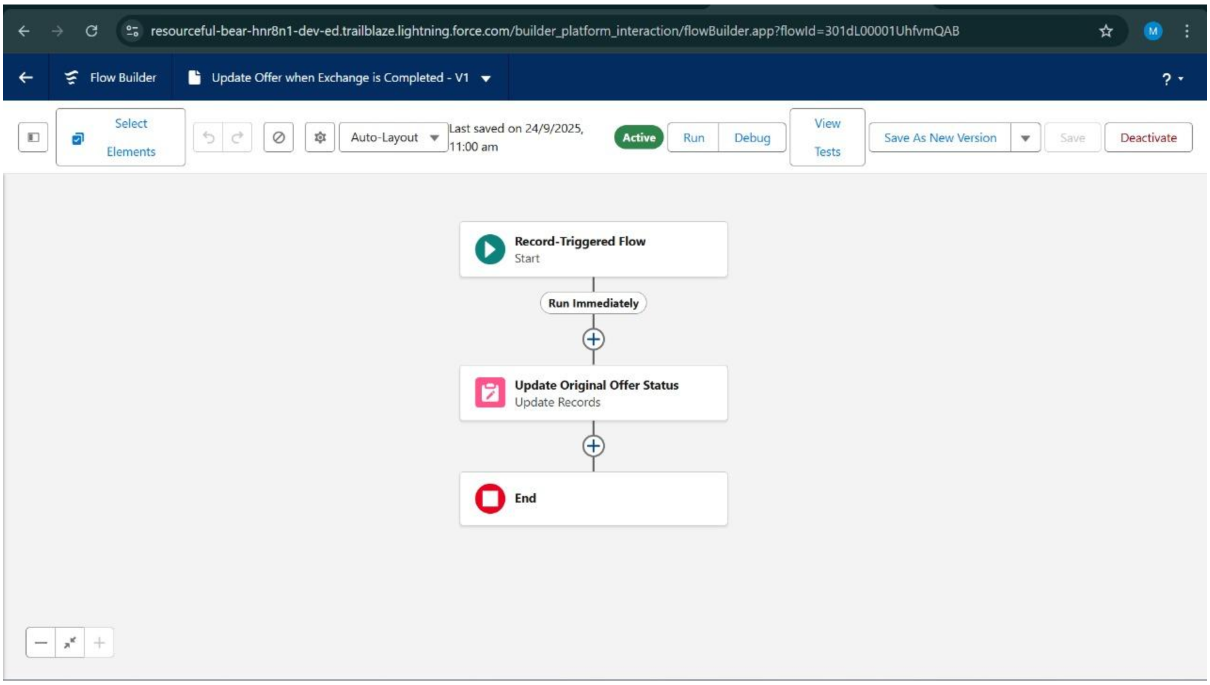
* Records Related to Exchange: Triggering Exchange__c > Original Offer

Multiple Offer__c records can be related to the Exchange triggering record. If there are no filter conditions, all related records are updated.

Set Filter Conditions

4.3 Final Flow and Activation

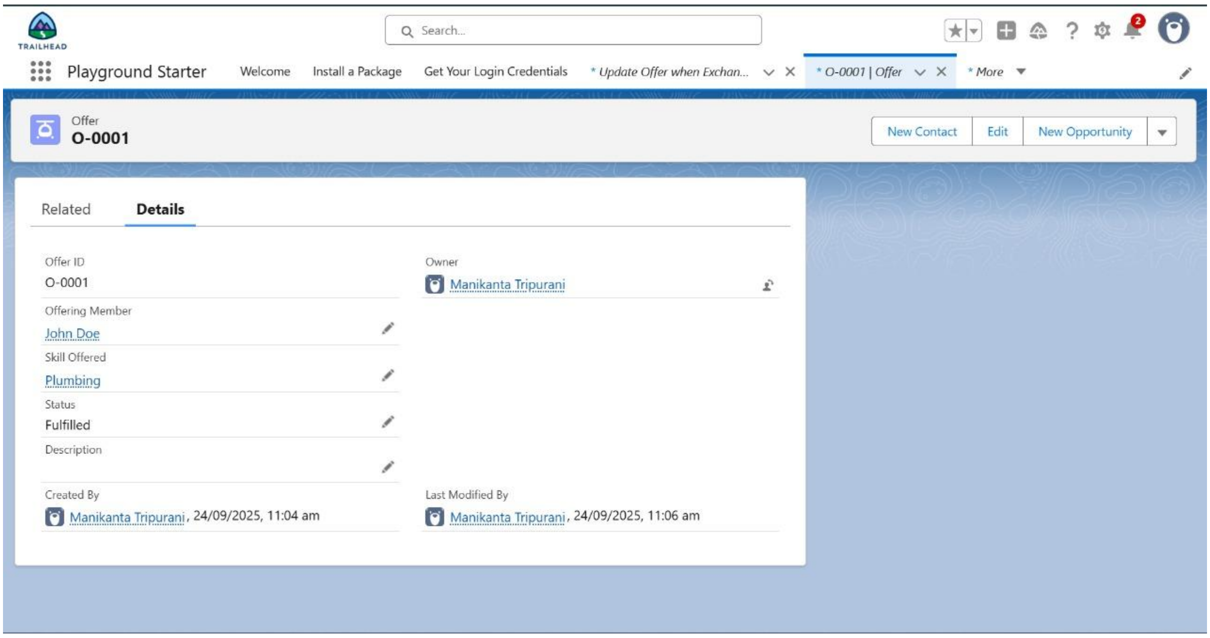
The complete Flow was saved with a descriptive label and activated, making it live in the organization. The final canvas shows the simple, efficient logic path from the trigger event to the final action, providing a clear visual representation of the automation.


resourceful-bear-hnr8n1-dev-ed.trailblaze.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301dL00001UhfvmQAB
Flow Builder
Update Offer when Exchange is Completed - V1
Select Elements
Auto-Layout
Active
Run
Debug
View Tests
Save As New Version
Save
Deactivate
Record-Triggered Flow
Start
Run Immediately
Update Original Offer Status
Update Records
End

Last saved on 24/9/2025, 11:00 am

5. Testing and Verification

The automation was tested by creating sample Offer and Exchange records.
The Exchange status was manually updated to "Completed," and we then verified that the Flow triggered correctly and updated the parent Offer record's status to "Fulfilled" as expected. The test was successful, confirming the automation works as designed.



Offer
O-0001

Offer ID
O-0001

Owner
Manikanta Tripurani

Offering Member
John Doe

Skill Offered
Plumbing

Status
Fulfilled

Description

Created By
Manikanta Tripurani, 24/09/2025, 11:04 am

Last Modified By
Manikanta Tripurani, 24/09/2025, 11:06 am

6. Expected Outcomes

- **Reduced Manual Effort:** Eliminates the need for NeighborNet administrators to manually update offer statuses.
- **Improved Data Accuracy:** Ensures data consistency between Exchange and Offer records automatically.
- **Streamlined Process:** Creates a more efficient and reliable process for managing the lifecycle of an offer.

7. Conclusion

Phase 4 successfully demonstrated the power of Salesforce Flow to automate critical business processes. By building and activating this Flow, we have made the NeighborNet application more intelligent and efficient. This foundational automation sets the stage for more complex logic that will be addressed with Apex programming in the next phase. The project is now ready to proceed to **Phase 5: Apex Programming**.