Designing Flows in the World of Low Code: A Salesforce Perspective

Kalpana Puli

kalpanapuli@gmail.com
Independent Researcher, Texas, USA

Abstract

The rise of low-code platforms has transformed the landscape of application development, enabling users with minimal coding experience to create robust applications. Salesforce, a leader in this domain, provides a suite of tools that facilitate the design and implementation of workflows through visual interfaces. This paper explores the principles of designing flows within Salesforce's low-code environment, highlighting the advantages, methodologies, and practical applications. By employing flowcharts, diagrams, and pseudocode, we aim to provide a comprehensive understanding of how businesses can leverage low-code solutions to enhance operational efficiency.

Keywords: Salesforce, Low-Code Development, Workflow Automation, Flow Design, Application Development

I . Introduction

In today's fast-paced digital environment, organizations are increasingly turning to low-code platforms to accelerate application development and streamline workflows. Salesforce stands out as a premier low-code platform, offering tools that empower users to build applications with minimal coding knowledge [1][2]. This paper delves into the design of flows within Salesforce's ecosystem, emphasizing the significance of low-code solutions in modern business operations.

Low-code development allows for rapid prototyping and deployment of applications, enabling organizations to respond swiftly to market demands. According to IDC, companies utilizing low-code platforms can deliver applications up to 68% faster than traditional methods [3]. This efficiency is crucial for maintaining competitiveness in an ever-evolving business landscape.

II.The Evolution of Low-Code Development

Historical Context

The concept of low-code development emerged in response to the growing demand for faster software delivery and the need for greater collaboration between IT and business teams. Traditional software development often involved lengthy coding processes that required specialized skills, leading to bottlenecks in project timelines. Low-code platforms democratize application development by allowing non-technical users (Business Users/Administrators) to participate actively in creating solutions.

Current Trends

Recent trends show that low-code platforms are increasingly popular across various industries, reflecting a significant shift in how organizations approach application development. A report from Gartner predicts that by 2025, 70% of new applications created by enterprises will leverage low-code or no-code technologies. This trend is driven by several key factors.

Many organizations are embarking on digital transformation initiatives, which emphasize the need for agility and innovation in their operations. As businesses strive to adapt to rapidly changing market conditions, the ability to quickly develop and deploy applications becomes essential. Low-code platforms enable faster development cycles, allowing companies to respond promptly to customer needs and market opportunities.

There is a notable shortage of skilled developers in the workforce. The demand for technical talent often exceeds the available supply, creating challenges for organizations looking to build robust software solutions. Low-code platforms help mitigate this issue by empowering non-technical usersoften referred to as citizen developersto create applications without extensive coding knowledge. This democratization of development not only alleviates pressure on IT teams but also encourages innovation from within the business and the cost efficiency associated with low-code development is a significant advantage.

By streamlining the development process and reducing the time required to bring applications to market, organizations can lower their overall costs while maintaining high standards of quality. This is particularly important in an economic environment where businesses are seeking to optimize resources and maximize return on investment.

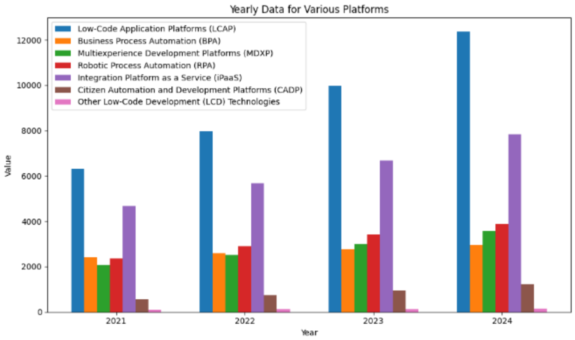


FIG II. report by Gartner [1]

III. Designing Flows in Salesforce

Understanding Flows

Salesforce Flows are powerful automation tools that allow users to create complex business processes without extensive coding [3]. Flows can be categorized into two main types: Screen Flows and Auto launched Flows. Screen Flows guide users through a series of screens for data input, while Auto launched Flows run in the background without user interaction.

Screen Flows: These flows guide users through a series of screens for data input.

Auto-Launched Flows: These run in the background, triggered by specific events without user interaction.

By combining these flow types, Salesforce provides businesses with the flexibility to design workflows that are both user-friendly and capable of automating complex tasks.

The Power of Flow Integration

One of the defining features of Salesforce Flow is its ability to seamlessly interact with external systems, enabling users to integrate third-party services and data sources without requiring extensive coding knowledge. This is accomplished through the capability of flows to directly call REST APIs from within the Salesforce platform, which allows organizations to leverage their existing systems while maintaining a streamlined development process.

By utilizing built-in actions such as the "Callout" element, users can easily send HTTP requests to external APIs, significantly simplifying the integration process. This approach eliminates the need for complex middleware or custom code deployments that are typically associated with traditional integration methods.

Consequently, businesses can automate workflows that span multiple applications, thereby minimizing technical debt and enhancing operational efficiency.

To effectively leverage flows for external API integration, organizations can follow several steps. First, they should identify external API requirements, determining which services need to be integrated and understanding their API specifications, including endpoints and authentication methods.

Next, users can create a new flow in Salesforce by navigating to Flow Builder and selecting either a Screen Flow or an Autolaunched Flow.

Once the flow is created, they can use the Callout element to specify the API endpoint and method (GET, POST, etc.), configuring any necessary headers and body content.

After setting up the callout, it is essential to handle API responses by establishing decision elements in the flow that manage various outcomes from the API calls such as success or error responsesand using variables to store returned data for further processing. Rigorous testing of the flow is crucial to ensure it interacts correctly with the external API and handles responses as expected.

Once testing is complete, organizations should deployand monitor the flow to ensure it meets business needs and performs effectively in a live environment. By following these steps, businesses can harness Salesforce Flow's capabilities to create efficient integrations with external APIs, facilitating improved automation and operational
effectiveness across theirprocesses.

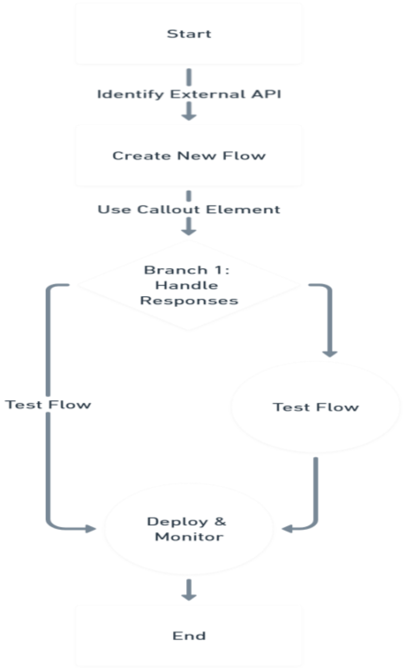


FIG III (a). how Salesforce Flows interact with external APIs[4]

Key Features of Salesforce Flows:

The Flow Builder is a visual interface that allows users to design and configure flows by simply dragging and dropping elements such as screens, decisions, loops, and actions. This intuitive design facilitates the creation of complex workflows without the need for extensive coding knowledge.

Salesforce Flows excel in dynamic data handling, enabling real-time reading and manipulation of Salesforce data. This capability allows businesses to create adaptive workflows that respond to changing conditions, ensuring that processes remain relevant and effective.

Process automation is a significant benefit of using Flows; repetitive tasks can be automated, which not only improves efficiency but also reduces the likelihood of manual errors.

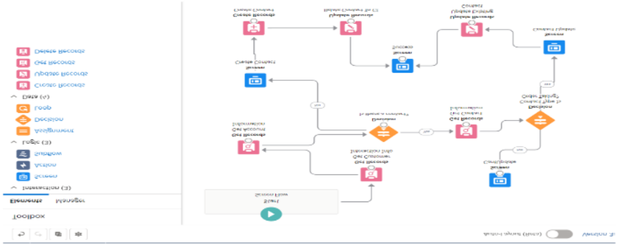


FIG III (b): Flow Builder[4]

Benefits of Salesforce Flows

Salesforce Flows offer numerous benefits that significantly enhance the development and management of business applications. One of the primary advantages is faster development; the low-code interface allows users to create applications and automate processes quickly, reducing the time from concept to deployment. This rapid development capability is crucial in today’s fast-paced business environment, where organizations must adapt swiftly to changing market conditions.

Salesforce Flows provide increased agility, enabling teams to implement changes to workflows with minimal effort. This flexibility allows businesses to respond promptly to evolving needs and capitalize on new opportunities without getting bogged down by lengthy development cycles.

The improved user experience that Salesforce Flows facilitate. The platform's intuitive visual interfaces empower users, even those with limited technical skills, to build and manage their workflows effectively. This democratization of application development not only enhances productivity but also encourages broader participation across departments, fostering a culture of innovation. By streamlining processes and making automation accessible, Salesforce Flows help organizations optimize their operations, improve collaboration, and ultimately drive better outcomes.

IV. Flow Design Principles

When designing flows in Salesforce, it is essential to adhere to several key principles to ensure effectiveness and usability. User-centric design is paramount; flows should be intuitive and easy for users to navigate, promoting a positive experience. Also Embracing modularity allows developers to break down complex processes into smaller, manageable components, making the overall flow easier to understand and maintain. This approach not only enhances clarity but also facilitates reusability across different flows.

Implementing robust error handling mechanisms is crucial as well; these mechanisms ensure that users are promptly informed of any issues that arise during execution, thereby improving the overall user experience.

The practice of testing and iteration is vital; regular testing of flows, combined with iterative improvements based on user feedback, helps refine functionality and adapt to changing business needs. By following these principles, organizations can create effective Salesforce flows that enhance operational efficiency and user satisfaction.

Below is a simplified flowchart illustrating a typical approval process using Salesforce Flow

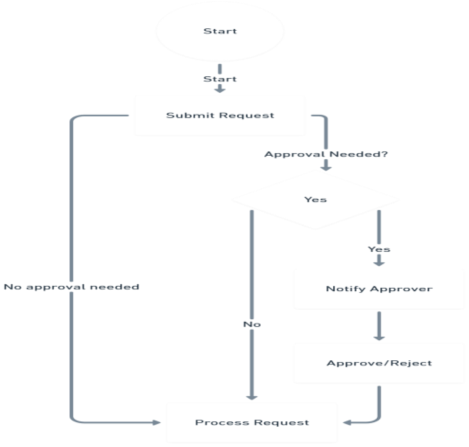


FIG IV: SalesForceWorkFlow

Pseudocode for Flow Logic

```
IF request is submitted THEN
    IF approval is needed THEN
        NOTIFY approver
    ELSE
        PROCESS request
    ENDIF
ENDIF
```

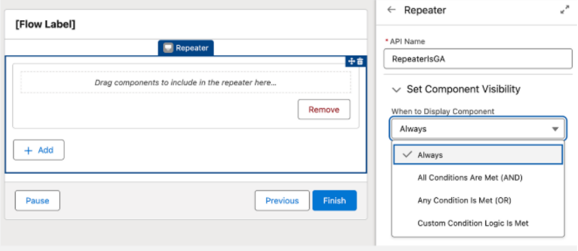
V. Flow Design Process in Salesforce

The process of designing effective flows in Salesforce begins with identifying the business flow needs. This initial step involves mapping out the business process by pinpointing key inputs, decision points, and outputs, which aids in structuring the flow appropriately. Utilizing flowcharts can be particularly beneficial, as they provide a visual representation of the process steps, making it easier to understand and communicate the workflow. Following this, the next phase is building the flow in Flow Builder, Salesforce's dedicated tool for flow creation. Users have the option to choose between screen-based or auto-launched flows and can incorporate various elements such as screens for data input, decision points for conditional logic, and loops for iterating through records. Additionally, actions can be added to create or update records, send emails, or invoke Apex code for more advanced custom logic. Once the flow is constructed, it is crucial to engage in testing and refining. This stage ensures that the flow operates as intended; Salesforce provides debugging tools that allow users to view real-time results and identify any issues. Refining the flow based on user feedback and performance metrics is essential to ensure that it remains efficient and user-friendly.

VI. Advanced Features in Flow Design

Salesforce continues to enhance its Flow capabilities with each release cycle. For instance:

Flow Repeater Component: Introduced in Summer '24 release allows users to collect information about multiple items from a single screen efficiently.



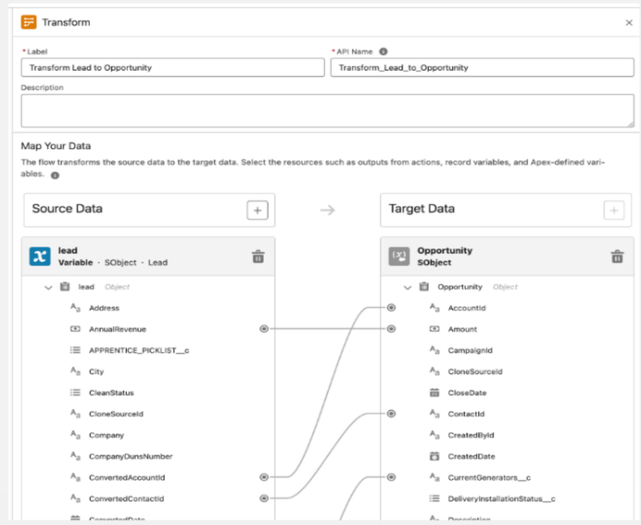
API Name: RepeaterIsGA

When to Display Component: Always

[Flow Label]
Repeater
Drag components to include in the repeater here...
Remove
+ Add
Pause
Previous
Finish
Repeater
Set Component Visibility
Always
All Conditions Are Met (AND)
Any Condition Is Met (OR)
Custom Condition Logic Is Met

FIG VI(a):Flow Repeater Component[8]

Transform Element: This feature enables users to modify data formats easily within flows without looping through records manually.



Label: Transform Lead to Opportunity

API Name: Transform_Lead_to_Opportunity

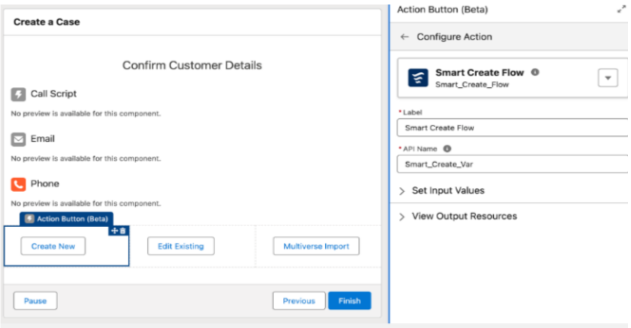
Description:

Map Your Data

The flow transforms the source data to the target data. Select the resources such as outputs from actions, record variables, and Apex-defined variables.

FIG VI(b): Transform Element [8]

Action Button Component: Users can now execute auto launched flows directly from a screen via an action button without navigating away from it .



Label: Smart Create Flow

API Name: Smart_Create_Var

No preview is available for this component.

FIG VI(c):Transform Element [8]

These advancements not only streamline workflow design but also empower citizen developers with more sophisticated tools at their disposal.

VII. Real-World Applications of Salesforce Flows

Case Study: Streamlining Customer Support Processes

One notable application of Salesforce Flows is in automating customer support processes.

A prominent insurance company effectively utilized Screen Flows to assist support agents in troubleshooting customer issues reported through various channels, such as phone calls and emails.

The implementation process began with the creation of a flow that guided agents by prompting them with specific questions related to the customer's issue, while simultaneously offering relevant troubleshooting steps based on their responses. To enhance the support experience, the flow was integrated with an internal knowledge base, allowing agents to access pertinent articles directly within the flow interface.

A feedback loop was established, enabling agents to provide input on the effectiveness of the troubleshooting steps after each interaction. This feedback was instrumental in refining and improving the flow over time, ensuring that it remained effective and aligned with user needs. By leveraging Salesforce Flows in this manner, the insurance company was able to streamline its customer support operations, enhance agent productivity, and ultimately improve customer satisfaction.

Results:

This approach led to a 30% reduction in average handling time per call while improving customer satisfaction scores significantly due to faster resolutions.

Case Study: Automating Sales Processes

Another compelling example of Salesforce Flows in action comes from a mid-sized manufacturing firm that implemented Auto-launched Flows to automate its lead qualification processes.

The first step in their implementation was to establish a lead capture integration, which triggered an Auto-launched Flow whenever new leads were submitted through their website forms. This flow was designed to automatically evaluate each lead against predefined criteria, such as budget range and company size, ensuring that only the most suitable leads were prioritized.

Following this assessment, the flow would assign qualified leads to sales representatives based on their availability, streamlining the handoff process.

The firm set up a notification system that sent alerts via email or SMS to sales team members once leads were qualified and assigned, facilitating prompt follow-up actions.

By leveraging Salesforce Flows in this manner, the manufacturing firm was able to enhance its lead qualification efficiency, reduce manual processing time, and improve overall sales productivity.

Results:

This automation resulted in improved lead response times by over 50%, allowing sales teams more time for personalized engagement rather than administrative tasks.

VIII. Challenges and Considerations

While Salesforce Flows offer significant advantages for organizations looking to enhance their operational frameworks, several challenges must be considered. One major challenge is complexity management; as flows become more intricate due to evolving business requirements or integrations with other systems, such as ERP solutions, it becomes essential to implement effective documentation practices to manage this complexity

There is a need for user training; despite the user-friendly design intended for non-developers and administrators, adequate training sessions are necessary to ensure that all users feel confident and capable of navigating these tools effectively. Furthermore, performance monitoring is critical post-deployment; organizations should conduct regular assessments not only to evaluate functionality but also to track performance metrics like execution times, as suboptimal performance can adversely affect the overall user experience. Addressing these challenges through careful planning and ongoing support will help organizations maximize the benefits of Salesforce Flows while minimizing potential pitfalls.

IX. Conclusion

The design of flows within Salesforce's low-code environment represents a significant advancement in how organizations approach application development and workflow automation. By leveraging user-friendly interfaces and powerful automation capabilities, businesses can enhance operational efficiency while empowering non-technical users to contribute meaningfully to software development efforts. Designing effective flows streamlines business processes while fostering innovation by allowing teams greater flexibility in responding to changing market demands. This exploration highlights how organizations can harness the power of low-code solutions like Salesforce Flows for improved operational performance while preparing for future advancements in this rapidly evolving field.

References

[1] Gartner forecasts worldwide low-code development technologies market to grow 20 percent in 2023. from https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023(accessed April. 10, 2023).
[2] Routine Automation, “Low-code application development with Salesforce.” Routine-Automation.comhttps://routine-automation.com/blog/low-code-application-development-with-salesforce/ (accessed June. 10, 2023).
[3] Salesforce, “What is Low Code? The complete development guide.” Salesforce.com. https://www.salesforce.com/in/platform/low-code-development-platform/what-is-low-code/ (accessed Nov. 10, 2023).
[4] Astraea IT Services, "Calling REST API from Flows in Salesforce." AstreaIT.com. https://astreait.com/Calling-REST-API-from-Flows-in-Salesforce/ (accessed July. 11, 2023).
[5] F. Sufi, “Algorithms in low-code-no-code for research applications: A practical review,” Algorithms, vol. 16, no. 2, p. 108, Feb. 2023, doi: 10.3390/a16020108.
[6] Forrester Research, “The Forrester Wave™: Low-Code Development Platforms for Application Development &Deployment Professionals.” Salesforce.com. https://www.salesforce.com/news/stories/forrester-names-salesforce-a-leader-among-low-code-app-development-platforms/ (accessed Aug. 12, 2023).
[7] Salesforce, “Introduction to Flow Part 1.” Salesforce Time. https://salesforcetime.com/2021/04/11/introduction-to-flow-part-1/ (accessed Nov. 12, 2023).
[8] Concret.io, “Top Salesforce Flow features in Summer '24 release.” Concret.io. https://www.concret.io/blog/top-salesforce-flow-features-in-summer-24-release (accessed Nov. 14, 2023).