

Phase 4: Automation

Automation is implemented to streamline processes, enforce business logic, and improve user experience. Key automations include Validation Rules and Flows.

Validation Rules

Validation rules ensure data integrity by preventing users from saving records with invalid data.

- **Example Rule:** Prevent Booking End Time Before Start Time
 - **Object:** Booking__c
 - **Rule Name:** Check_Booking_End_Time
 - **Error Condition Formula:** End_Time__c < Start_Time__c (Only applicable if End_Time__c is populated)
 - **Error Message:** "Booking End Time cannot be earlier than the Start Time."
 - **Purpose:** Ensures logical consistency in booking timeframes.

The screenshot shows the Salesforce Object Manager interface for the 'Booking' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Validation Rules' and displays two items, both named 'Booking_End_After_Start'. The first rule has an error message 'End time must be after start time.' and was modified by 'Uppari Pedda Venkatesh' on 10/16/2025 at 12:35 AM. The second rule has an identical error message and was modified by the same user on 10/17/2025 at 3:10 AM. The interface includes a top navigation bar with tabs for Recently Viewed, Accounts, Booking, and Charging Slot, along with standard browser controls and a status bar at the bottom.

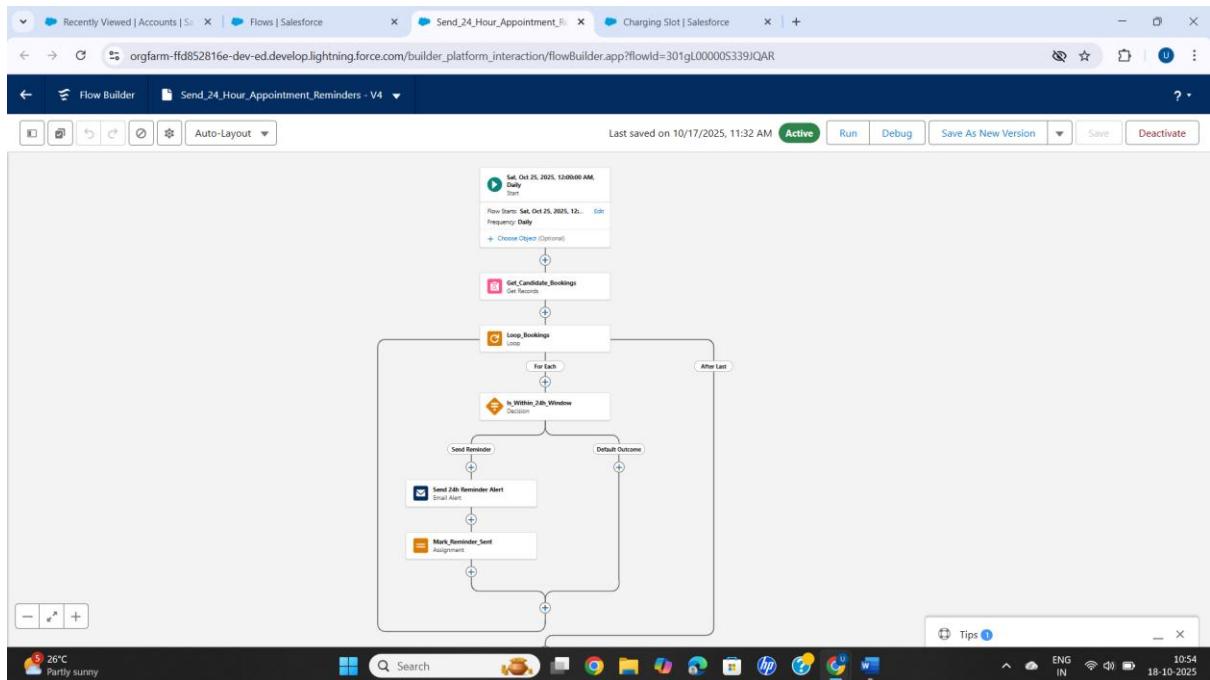
Rule Name	Error Location	Error Message	Active	Modified By
Booking_End_After_Start	Top of Page	"End time must be after start time."	✓	Uppari Pedda Venkatesh, 10/16/2025, 12:35 AM
End_After_Start	Top of Page	End time must be after Start time.	✓	Uppari Pedda Venkatesh, 10/17/2025, 3:10 AM

Flows

Flows automate complex business processes, including user interactions, record updates, and scheduled actions.

- **Scheduled Flow: Send 24-Hour Appointment Reminders** 

 - **Flow Type:** Schedule-Triggered Flow.
 - **Trigger:** Runs on a defined schedule (e.g., Hourly). Configured to start at a specific time and repeat every hour. Does not require an object trigger, it runs purely based on time.
 - **Logic:**
 1. **Get Records:** Finds Booking__c records where the Start_Time__c field is exactly 24 hours greater than the current time (`$Flow.CurrentDateTime + (1/24)`) AND the Reminder_Sent__c checkbox is false. Stores all found records in a collection variable.
 2. **Loop:** Iterates through the collection of Booking records found in the previous step.
 3. **Send Email Alert:** Inside the loop, uses the standard **Email Alert** action. Selects the pre-configured BookingReminder_24h_EmailAlert. Passes the current Booking record's ID (`{!LoopVariable.Id}`) to the action so it knows which record's details (like Contact email) to use.
 4. **Assignment (Add to Collection):** Inside the loop, adds the current Booking record (specifically, its ID and the Reminder_Sent__c field set to true) to a *new* record collection variable intended for updates.
 5. **Update Records:** After the loop finishes, uses a single **Update Records** element to update all the Booking records stored in the update collection (from step 4), setting their Reminder_Sent__c field to true. This bulk update is efficient and avoids hitting limits.

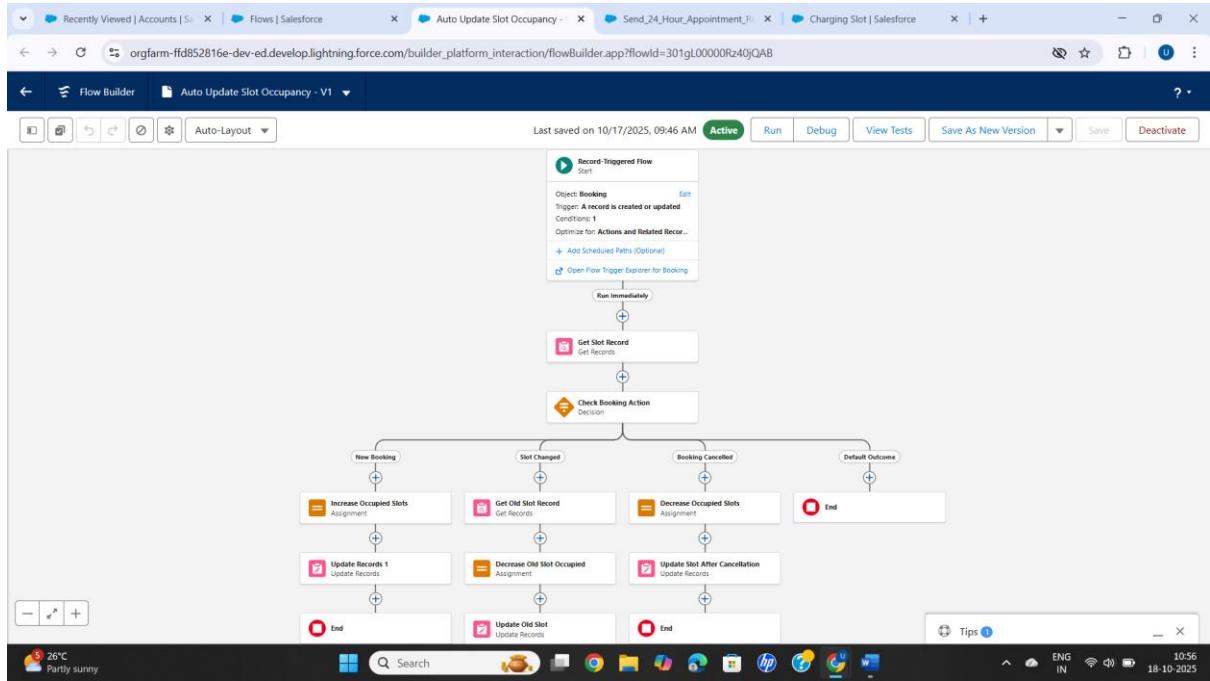


Record-Triggered Flow: Auto Update Slot Occupancy (Initial Version) [🔗](#)

- **Flow Type:** Record-Triggered Flow.
- **Object:** Booking__c.
- **Trigger:** Configured to run *after* the record is saved (to ensure the Booking status is final). Runs on both **Create** and **Update**.
- **Entry Conditions:** Optimized to run only when Status__c Is Changed true. Further refined with formula condition: OR(ISPICKVAL({!\$Record.Status__c}, 'Confirmed'), ISPICKVAL({!\$Record.Status__c}, 'Cancelled')).
- **Logic:**
 1. **Get Records:** Retrieves the *single* related Slot__c record where Id equals {!\$Record.Slot__c} (the Slot lookup on the triggering Booking).
 2. **Decision:** Checks the value of the triggering Booking's Status__c field.
 - Outcome 1: Status is Confirmed.
 - Outcome 2: Status is Cancelled.

3. Update Records (Confirmed Path): Updates the retrieved Slot__c record, setting its Availability__c field to 'Unavailable'.

4. Update Records (Cancelled Path): Updates the retrieved Slot__c record, setting its Availability__c field to 'Available'.



Screen Flow: Guided Station Creation 🌟

- Flow Type:** Screen Flow.
- Launch:** Could be launched from a button on the Account page or a dedicated tab.
- Logic:**
 - Screen 1:** Collects Station details (Name, Address using Address component, Status).
 - Create Records:** Creates the Station__c record using the inputs from Screen 1.
 - Screen 2:** Asks how many Slots to create for this Station (e.g., using a Number input).
 - Loop:** Loops from 1 to the number entered in Screen 2.

5. **Assignment (Inside Loop):** Creates a *single* Slot record variable, setting Station__c to the ID created in step 2, Name (e.g., Station Name + Loop Counter), Availability__c to 'Available', and default Hourly_Rate__c.
6. **Assignment (Add to Collection - Inside Loop):** Adds the single Slot variable (from step 5) to a Slot *collection* variable.
7. **Create Records:** After the loop, creates all the Slot records stored in the collection variable (from step 6) in one go.
8. **Screen 3:** Confirmation message displaying the new Station name and number of slots created.

