**Phase 3: Data Modeling & Relationships**

**➣Standard & Custom Objects**

For this project, the standard Contact object is used to represent citizens. Contacts are extended with additional fields such as Zone and Reward Points to capture recycling behavior. Alongside this, several custom objects were created:

**• Waste Bin (Waste\_Bin\_\_c):** Stores bin location, fill level, and IoT sensor details.

**• Truck (Truck\_\_c):** Represents collection vehicles, drivers, and capacity.

**• Route (Route\_\_c):** Defines service areas and schedules.

**• Pickup Request (Pickup\_Request\_\_c):** Created whenever a bin requires collection.

**• Recycling Record (Recycling\_Record\_\_c):** Logs citizen recycling contributions and calculates reward points.

**➣Fields**

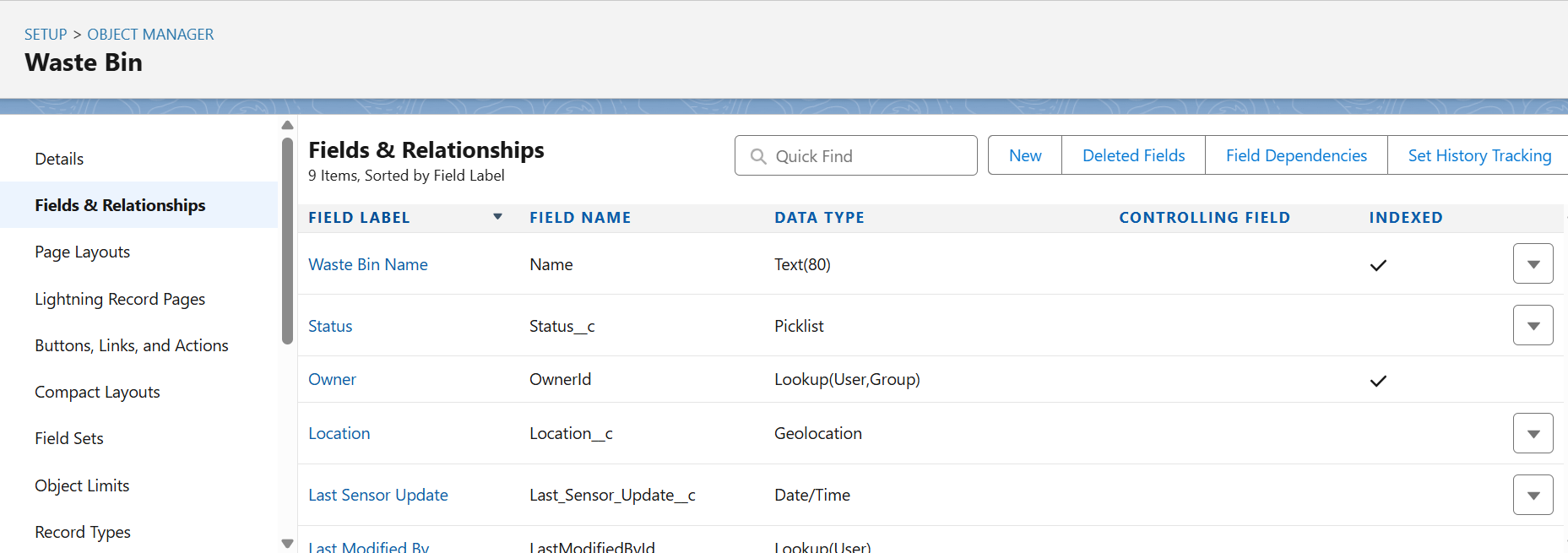
Custom fields were added to capture project-specific data:

**• Waste Bin:** Bin ID (Text), Location (Geolocation), Fill Level (Number), Status (Picklist).

**• Truck:** Truck ID (Text), Driver (Lookup to User), Capacity (Number), Current Load (Number), Status (Picklist).

**• Pickup Request:** Auto Number, Bin (Lookup), Assigned Truck (Lookup), Status (Picklist), Priority (Picklist), Scheduled Time (DateTime).

**• Recycling Record:** Auto Number, Citizen (Master-Detail), Material Type (Picklist), Quantity (Number), Points Earned (Number).



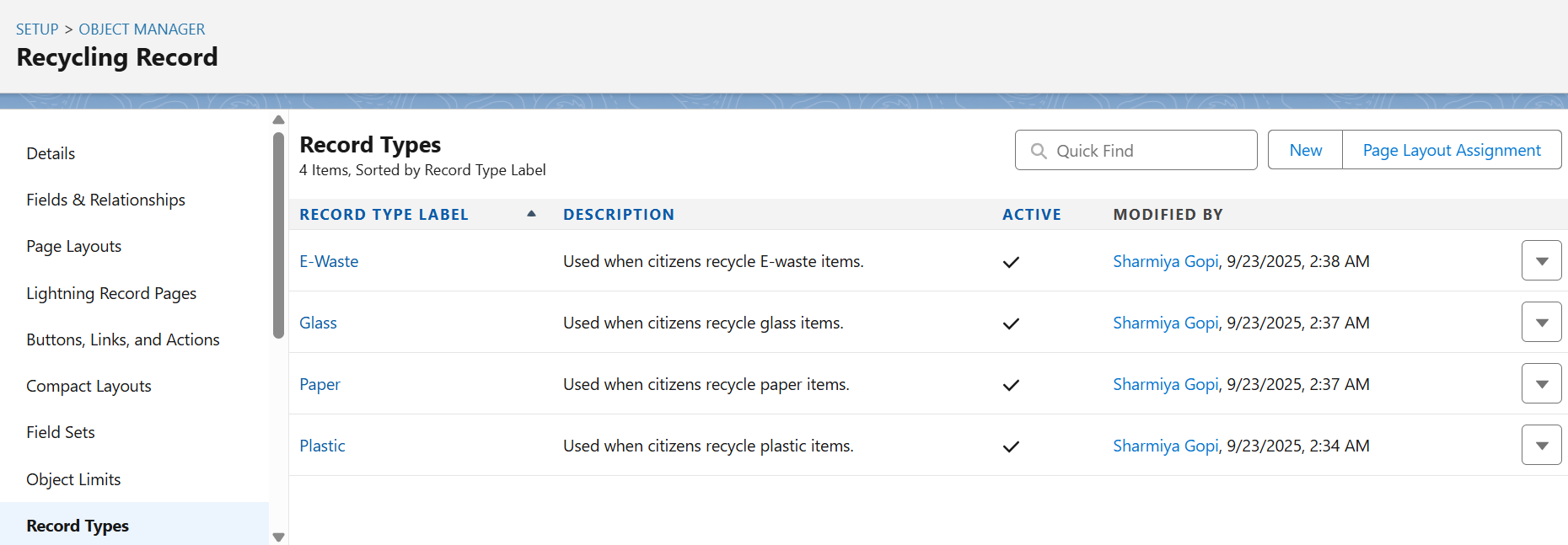
**➣Record Types**

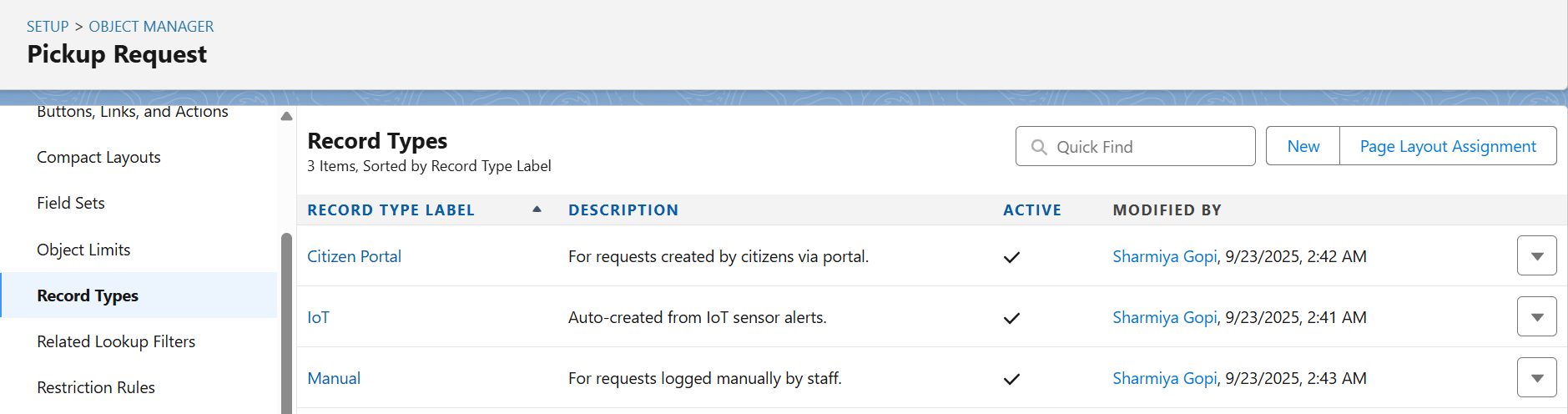
Record types were introduced to support process variations:

**• Pickup Request:** IoT, Citizen Portal, Manual.

**• Recycling Record:** Plastic, Paper, Glass, E-Waste.

Each record type uses different page layouts to tailor fields for the process.





**➣Page Layouts**

Page layouts were customized per object to group fields logically:

**• Waste Bin:** Bin details, sensor readings, collection history.

**• Truck:** Vehicle information, assigned driver, capacity tracking.

**• Pickup Request:** Request details, assigned truck/driver, scheduling.

**• Recycling Record:** Citizen details, material type, quantity, and points.

Related lists (e.g., Pickup Requests on Bin, Recycling Records on Contact) were also included.

**➣Compact Layouts**

Compact layouts were defined to optimize the mobile and record header view:

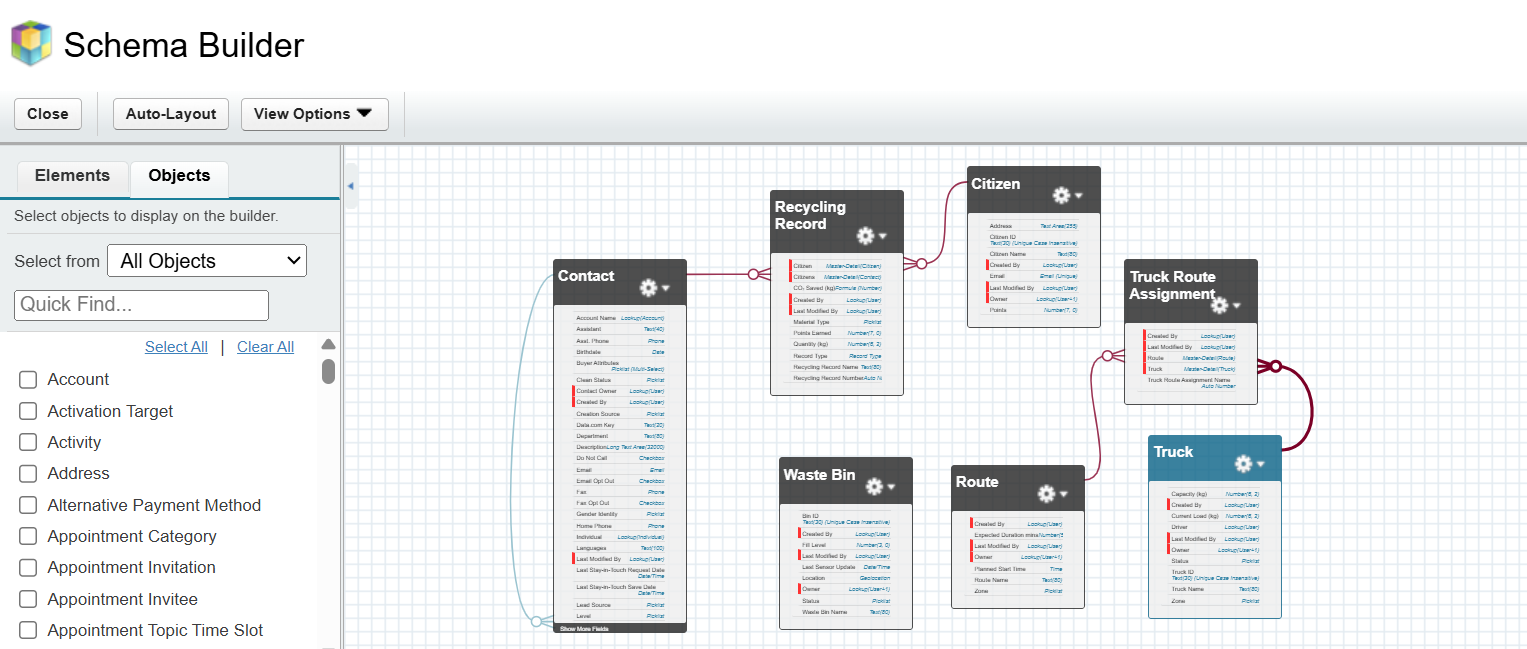
**•** Waste Bin Compact: Bin Name, Fill Level, Status.

**•** Pickup Request Compact: Request Number, Status, Assigned Truck.

**•** Recycling Record Compact: Record Number, Points Earned, Material Type.

**➣Schema Builder**

The Schema Builder was used to visualize objects, fields, and relationships. Custom objects (Waste Bin, Truck, Route, Pickup Request, Recycling Record) were added to the canvas to verify relationships. This provided a clear diagram for validation and documentation.



**➣Lookup vs Master-Detail vs Hierarchical Relationships**

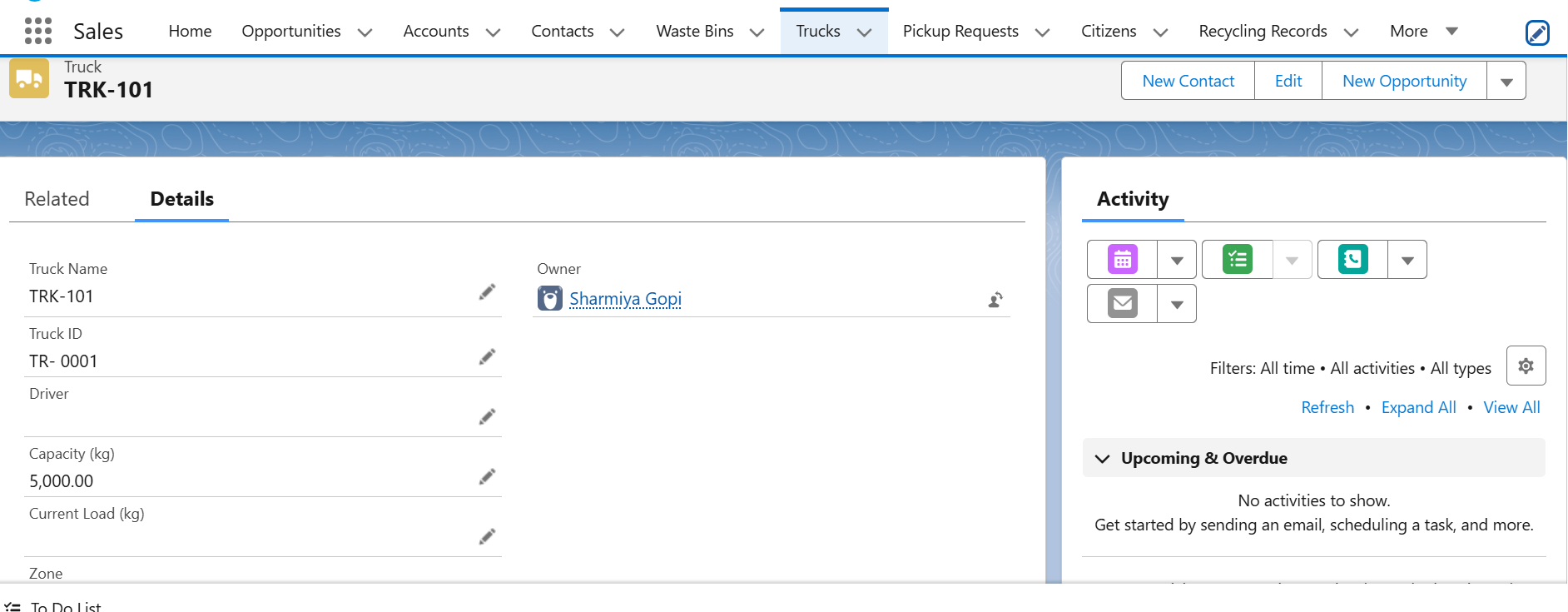
**• Master-Detail:** Used where strict dependency and roll-up summaries are required (e.g., Recycling Record → Contact, Truck Route Assignment → Truck and Route).

**• Lookup:** Applied where records are independent but linked (e.g., Pickup Request →Waste Bin, Pickup Request → Truck).

**• Hierarchical:** Salesforce only supports this on the User object and was not required in this project.

## **➣**Junction Objects

## A junction object, **Truck Route Assignment**, was created to model the many-to-many relationship between trucks and routes. This allows one truck to serve multiple routes and one route to have multiple trucks assigned.



**Phase 4: Process Automation (Admin)**

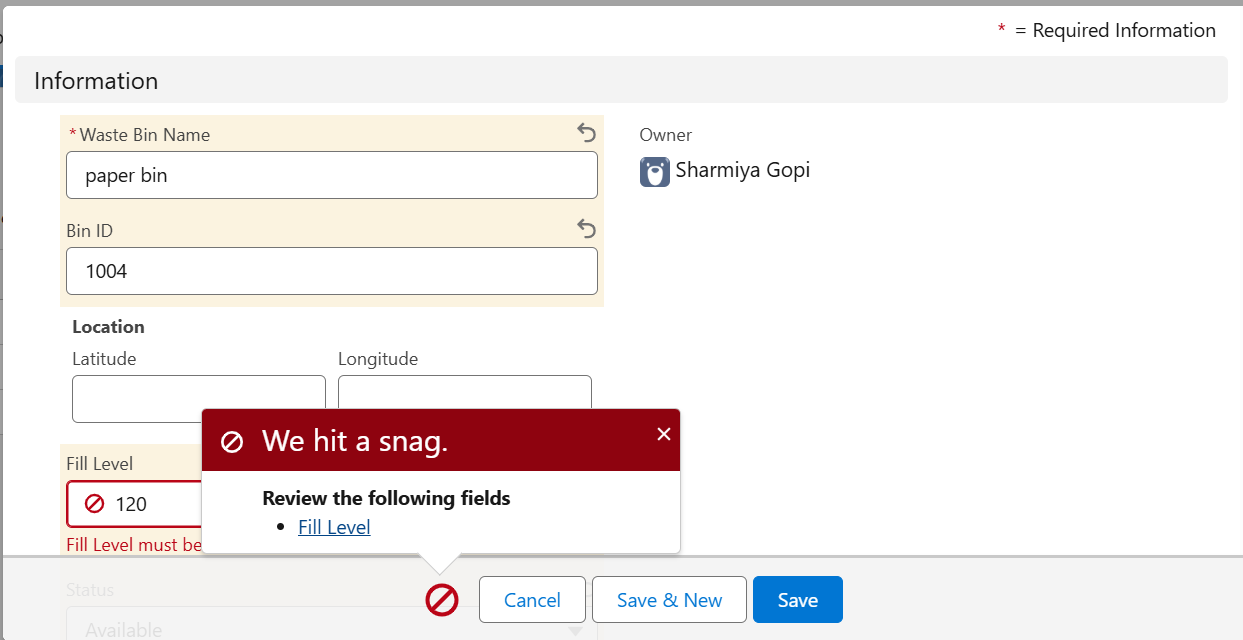
**➣Validation Rules**

Validation rules were used to enforce business rules and maintain data quality:

**• Truck Capacity Rule:** Prevents assigning a pickup request to a truck if the combined load would exceed its maximum capacity.

**• Bin Fill Level Rule:** Ensures IoT fill levels are always between 0% and 100%.

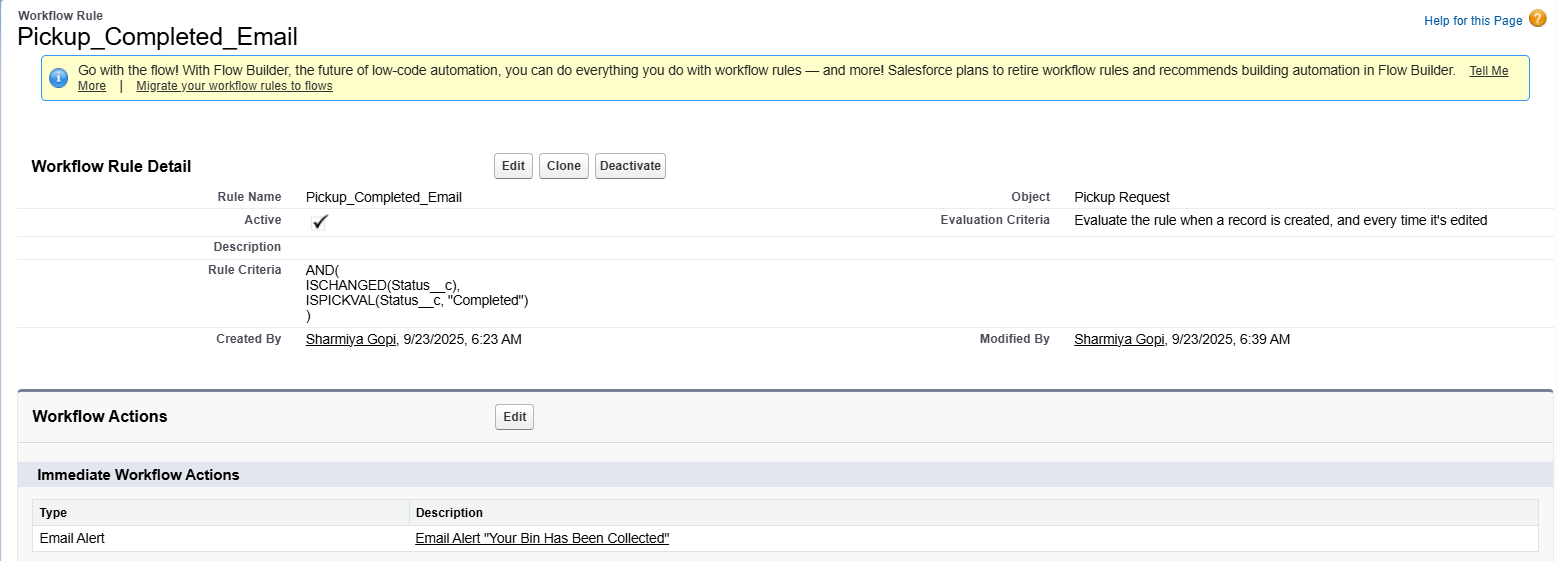
These rules reduce errors and prevent unrealistic data entry.



**➣Workflow Rules**

Although Workflow is legacy, a simple rule was created for demonstration:

**• Pickup Completion Notification:** When a pickup request status is changed to Completed, an automated email is sent to the citizen confirming the service.

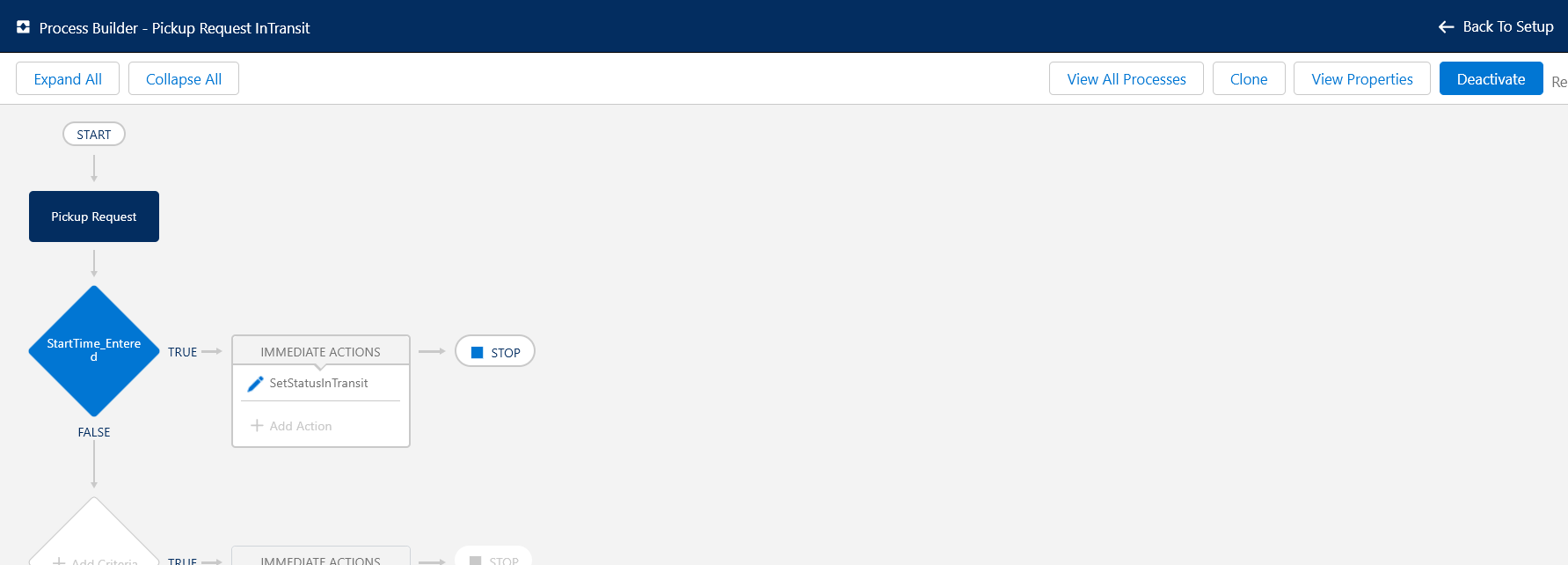


**➣Process Builder**

Process Builder was applied for lightweight updates before migrating logic to Flow:

**• Pickup Request In-Transit:** Automatically updates the pickup request status to In Transit once a driver logs a start time.

This showcased Salesforce automation prior to Flow Builder adoption.



**➣Approval Process**

An approval process was configured to handle escalation scenarios:

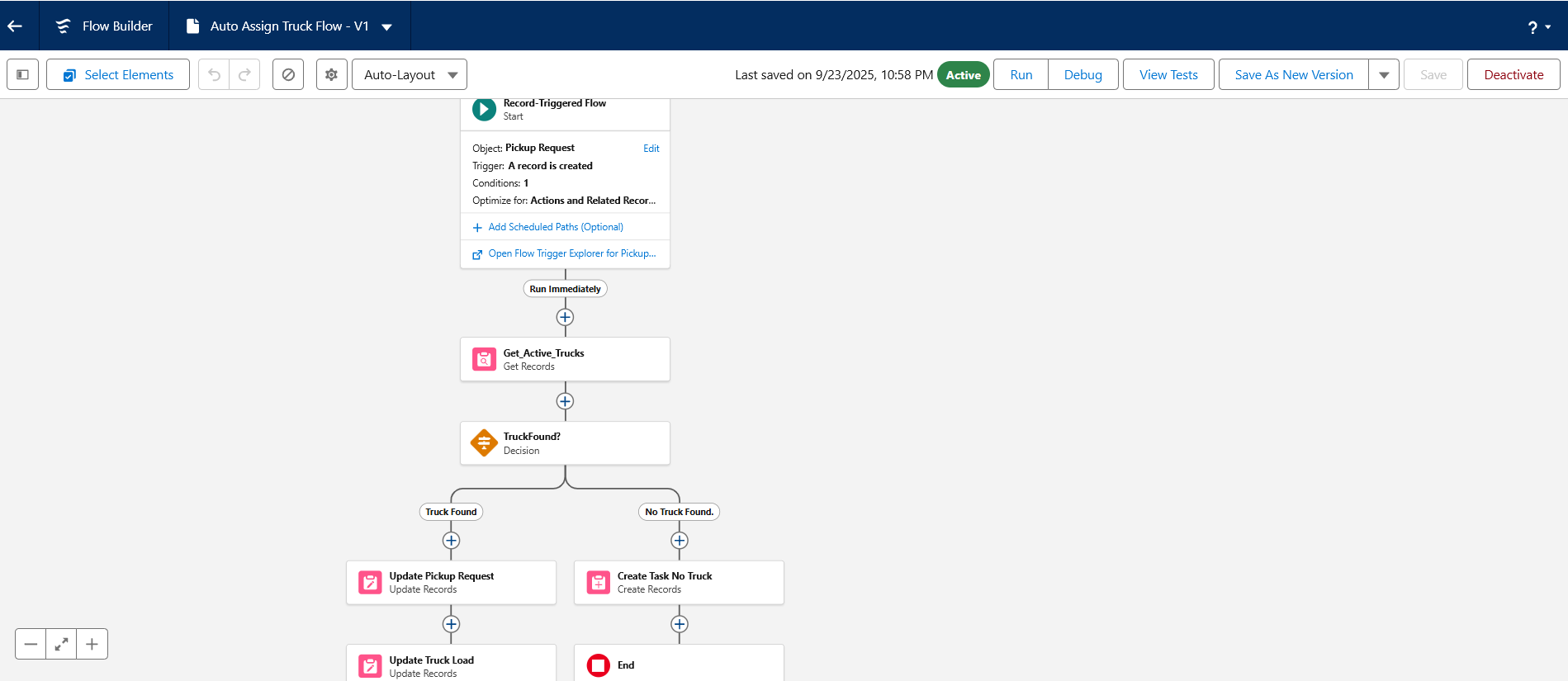
**• Emergency Pickup Approval:** If a pickup request is marked Emergency or the estimated load exceeds a threshold, it requires approval from the Commissioner before dispatch.

This ensures high-impact cases are reviewed by leadership.

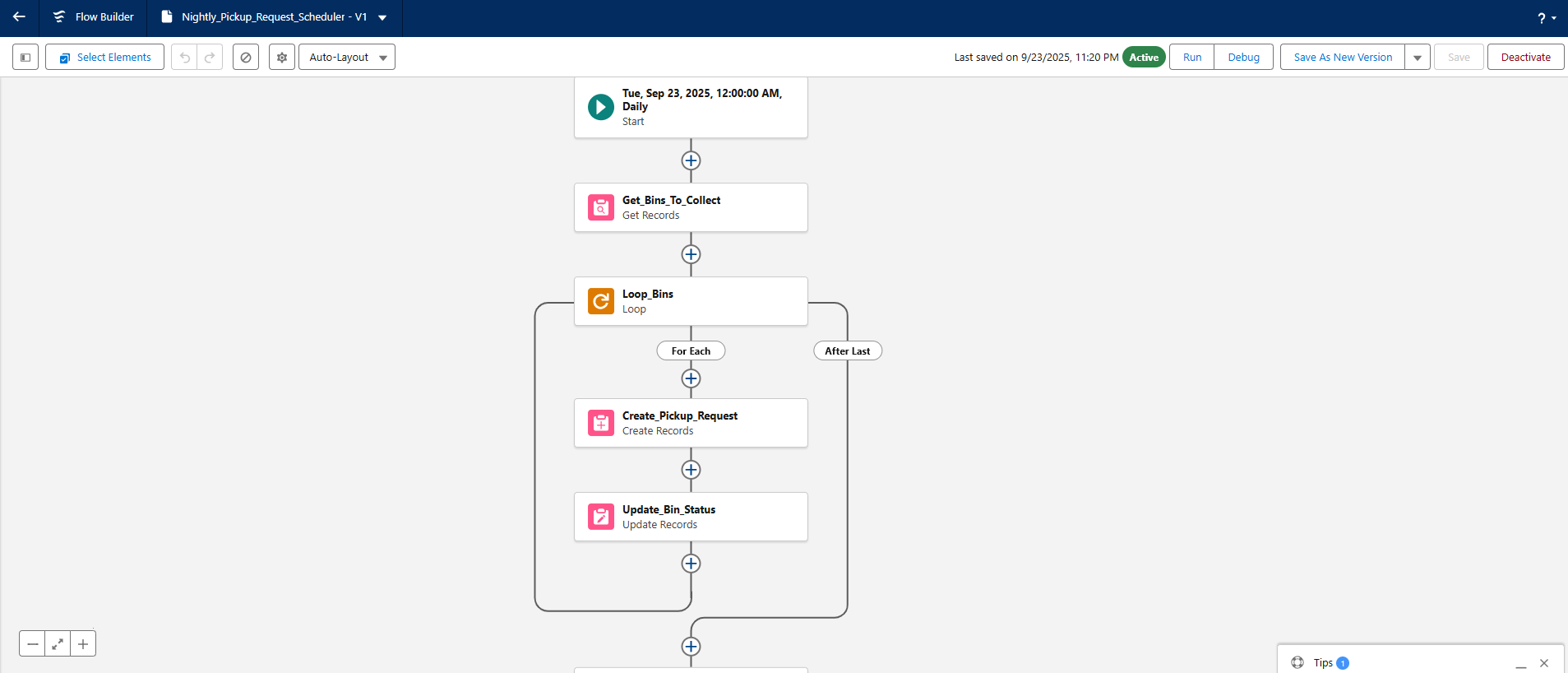
**➣Flow Builder**

Flow Builder was the core automation tool, covering multiple scenarios:

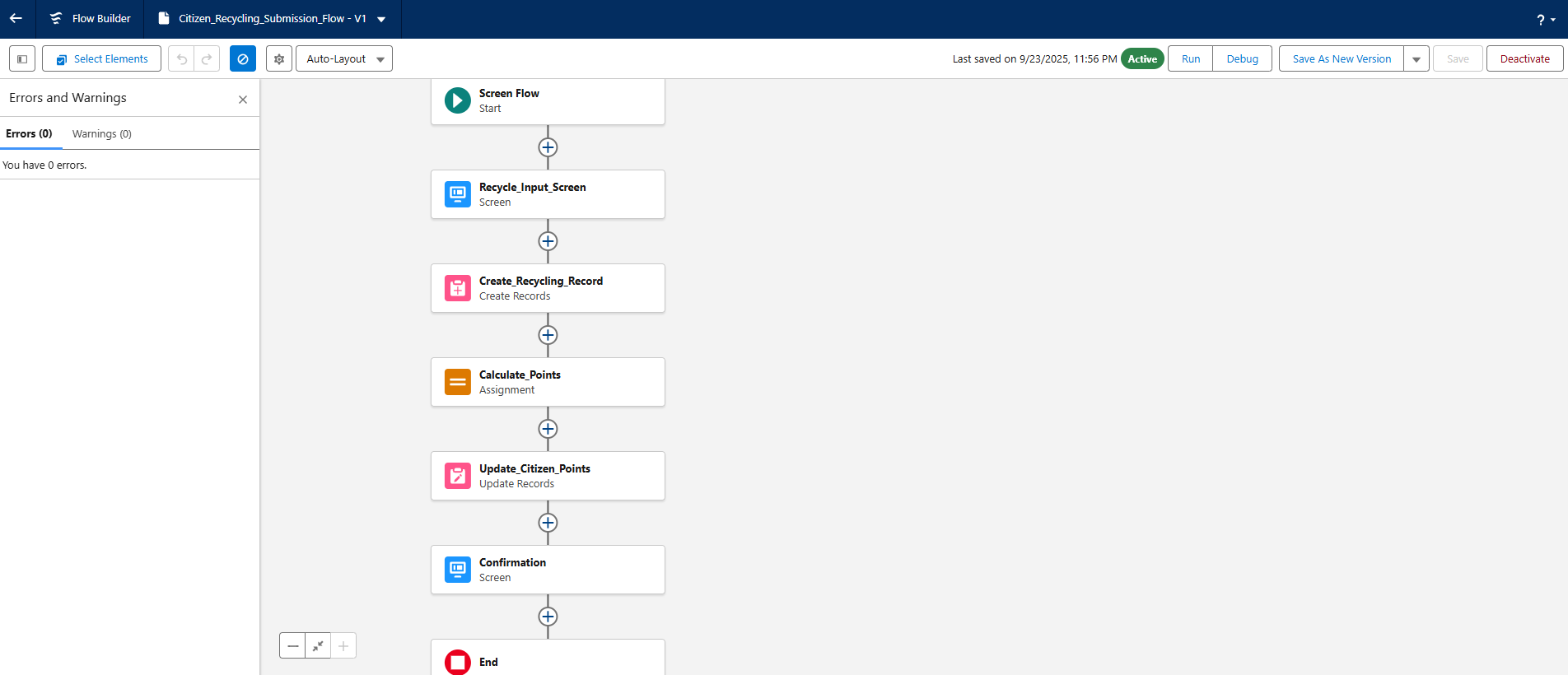
**• Record-Triggered Flow:** Auto-assigns the nearest available truck when a pickup request is created, updates the truck’s load, and notifies the driver.



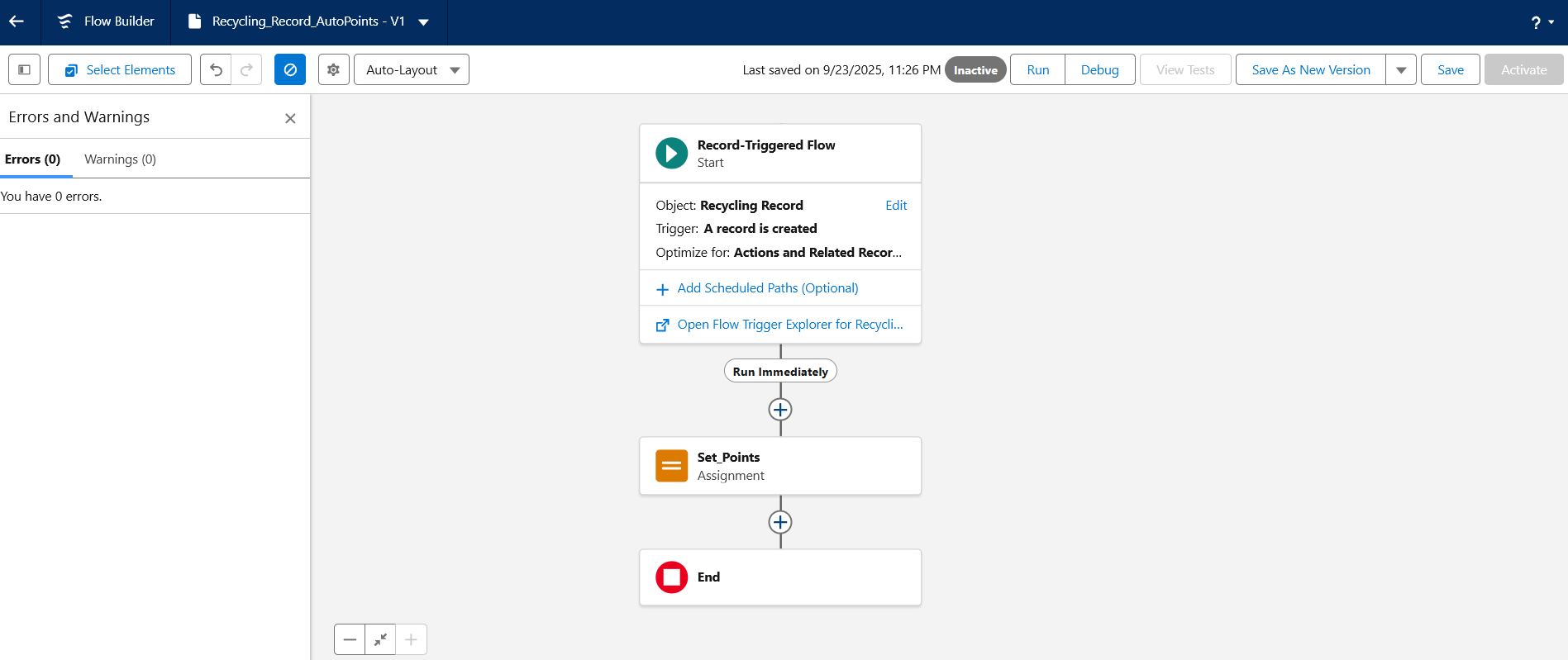
**• Scheduled Flow:** Runs nightly to check for bins with fill level above 80% and auto-creates pending pickup requests.



**• Screen Flow:** Embedded in the Citizen Portal to allow residents to submit recycling records and instantly update their reward points.



**• Auto-launched Flow:** Used for backend calculations, such as awarding recycling points when a record is logged.



Flows reduced the need for custom code while handling complex automations.

**➣Email Alerts**

Email alerts were linked with workflow rules and flows:

**•** Citizens received confirmation when their bin was collected.

**•** Drivers and supervisors received alerts for newly assigned pickups.

**➣Field Updates**

Automatic updates kept data synchronized:

**• Recycling Record:** Auto-calculates reward points based on material and quantity.

**• Contact (Citizen):** Updates cumulative reward points whenever a new recycling record is added.

**➣Tasks**

Automated task creation was used for driver reminders:

**•** When a pickup was assigned, a task was created for the driver with due date and bin details.

This ensured accountability and scheduling support.

**➣Custom Notifications**

Custom in-app/mobile notifications were built using Notification Builder:

**• Driver Assignment Notification:** Sent instantly when a truck driver is assigned a new pickup request.

**• Operations Alert:** Triggered if no trucks are available for a pending pickup request.

These real-time notifications improved responsiveness for staff and citizens.