

Metro Ticket Booking Automation – ServiceNow

Testing & Validation

Purpose

The testing and validation phase ensures that the **Metro Ticket Booking Service Catalog item**, **Flow Designer automation**, **data storage**, and **user notifications** work correctly according to defined requirements.

Testing validates the end-to-end flow from **ticket booking submission** to **database record creation**, ensuring accuracy, reliability, and smooth user experience.

Test Environment

- **Platform:** ServiceNow
 - **Modules Used:**
 - Service Catalog
 - Flow Designer
 - Custom Metro Database Table
 - **User Roles:**
 - Passenger (Requester)
 - System Administrator
-

Test Scenarios Covered

- Metro ticket booking submission
- Smart card recharge details capture

- Flow Designer trigger execution
- Metro database record creation

Testing Execution Details

Scenario 1: Metro Ticket Booking Submission

- User selects **Book A Metro Ticket Item** from the Service Catalog
- User enters required details:
 - Smart Card Number
 - Smart Card Holder Name
 - Starting Station
 - Destination Station
 - Number of Passengers
 - Journey Type
 - Mode of Payment
- User submits the request successfully

Figure 1: Book A Metro Ticket catalog item form

Scenario 2: Flow Designer Automation Trigger

- Flow is triggered automatically using **Catalog Item Requested** event
- Flow action retrieves all catalog variables

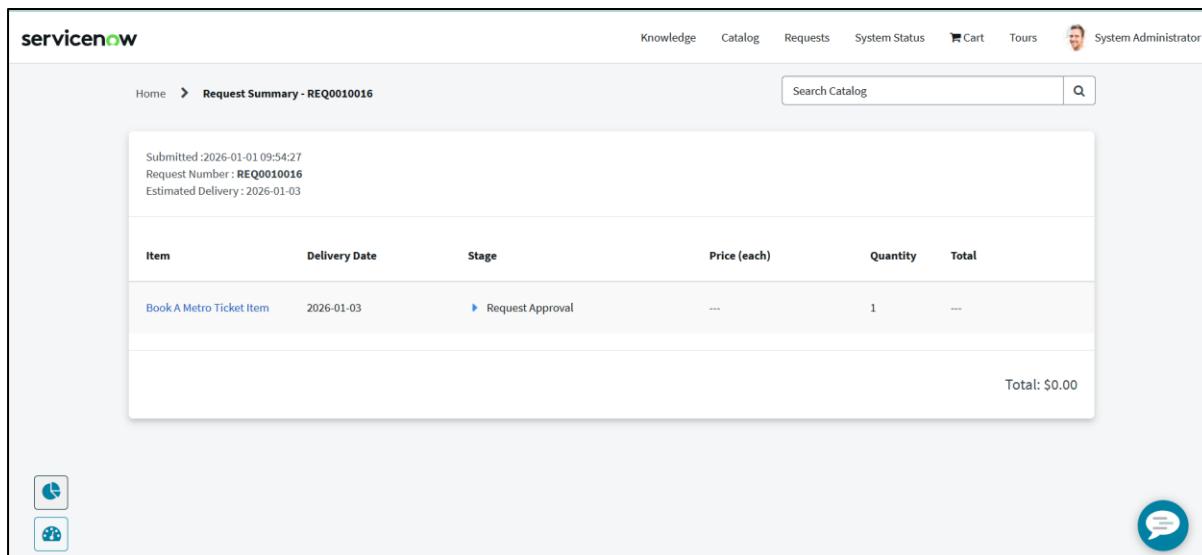


Figure 2: Metro Ticket submission

Scenario 3: Metro Database Record Creation

- Flow creates a record in the **Metro Database table (u_metro_database)**
- Fields populated include:
 - Smart Card Number
 - Card Holder Name
 - Recharge Amount
 - Starting From
 - Going To
 - Passenger Count

- Mode of Payment
- User Details
- Created Date

The screenshot shows a database table with the following columns: User Details, Smart Card Number, Smart Card Name, Recharge Amount, Mode of Payment, and Created. The data includes entries for users Boomika, Nancy, and RUPAY, along with their respective card numbers, recharge amounts (e.g., 123456789, 114144, HYD90109YTD), payment modes (UPI), and creation dates (e.g., 2025-12-31 08:32:46, just now, 2026-01-01 09:58:21).

User Details	Smart Card Number	Smart Card Name	Recharge Amount	Mode of Payment	Created
(empty)	Boomika	123456789	999	UPI	2025-12-31 08:32:46
(empty)	Boomika	114144	126	UPI	2025-12-3 just now
(empty)	Nancy	112432	1,000	UPI	2026-01-01 09:58:21
(empty)	Boomika	114144	126	UPI	2026-01-01 09:55:09
(empty)	(empty)			UPI	2025-12-31 08:30:49
(empty)	RUPAY	HYD90109YTD	1,000	UPI	2025-12-31 08:36:14

Figure 3: Metro Database table showing stored ticket booking details

Scenario 4: Flow Execution Validation

- Flow tested using **Test Run** option in Workflow Studio
- Execution completed without errors
- Both actions executed successfully:
 - Get Catalog Variables
 - Create Metro Database Record

The screenshot displays the execution details for a flow named 'Metro Project'. It shows the following information:

- EXECUTION DETAILS:** Metro Project
- FLOW STATISTICS:** Run as: System Administrator, Open flow logs, State: Completed, Start time: 2026-01-01 09:55:09, Duration: 25ms.
- TRIGGER:** Catalog Item Requested.
- ACTIONS:**
 - 1. Get Catalog Variables from Book A Metro Ticket Item (Core Action, Completed, 2026-01-01 09:55:09, 13ms)
 - 2. Create Record (Core Action, Completed, 2026-01-01 09:55:09, 9ms)
- ERROR HANDLER:** None.

Figure 4: Flow execution details showing successful completion