Metro Ticket Generating System
Testing and Validation

Introduction

The testing phase validates that the **Metro Ticket Generating System**, including the Service Catalog, Flow Designer automation, and digital QR fulfilment, functions correctly according to business requirements. Testing was performed in a controlled ServiceNow environment using valid test data to ensure all core functional paths specifically station selection, fare calculation, and QR rendering were verified end-to-end.

Test Environment

The validation was conducted using the following environment configurations:

- **Platform**: ServiceNow.
- **Modules Tested**: Service Catalog and Flow Designer.
- **User Roles Involved**:
    - **Commuter (Requester)**: Initiates the metro ticket booking or smart card recharge.
    - **IT Administrator**: Manages the backend data records and automation logic.

Test Execution Details

Testing focused on verifying that user inputs were accurately captured, fares were calculated dynamically, and records were stored securely without manual intervention.

Field Validation and Automation

- **Mandatory Fields**: Enforced on both the client and server side to prevent incomplete submissions, such as booking without a source or destination.
- **Auto-Populate Logic**: Catalog client scripts and UI policies correctly set values for single or return journeys and calculated the final fare in real-time.
- **Field Mapping**: Used the Process Automation Engine to ensure that values entered by commuters in the catalog variables were perfectly mapped to the corresponding fields in the custom database.

User Interface (UI) Validation Screenshots

The following images validate the dynamic behavior of the Service Catalog form based on user selection.

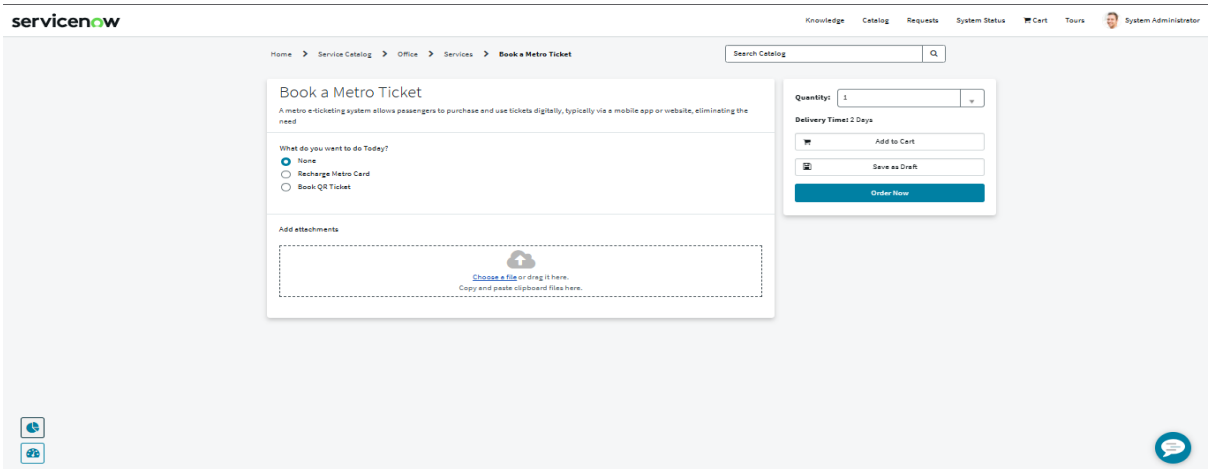


Figure 1: Default Form State (None Selected)

This view shows the initial state of the form where only the primary selection variable is visible to reduce initial clutter.

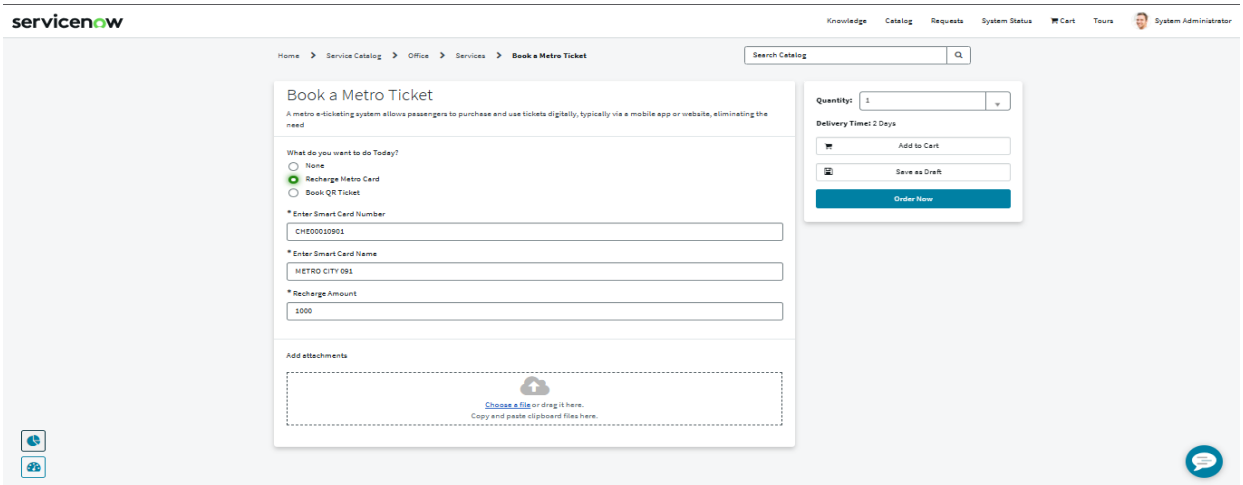


Figure 2: Smart Card Recharge Selection

When "Recharge Metro Card" is selected, the form dynamically reveals card-specific variables like Smart Card Number and Name.

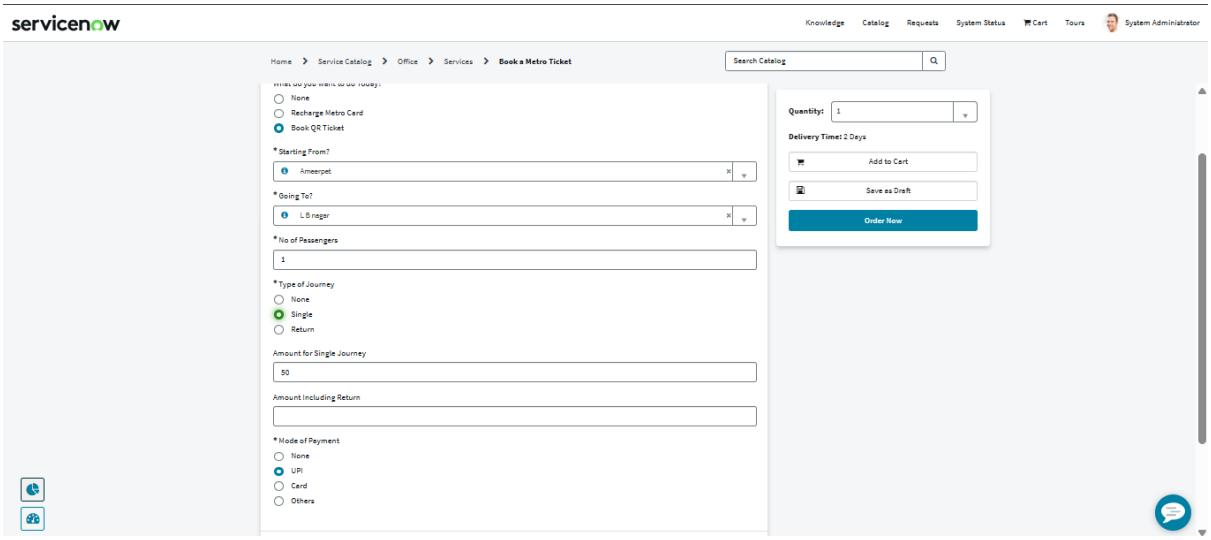


Figure 3: QR Ticket Booking Selection

When **Book QR Ticket** is selected, the form reveals journey station references and the auto-calculated fare preview.

Workflow and QR Execution

1. **Workflow Trigger**: The Flow Designer logic was successfully triggered automatically immediately following the submission of a new ticket request.

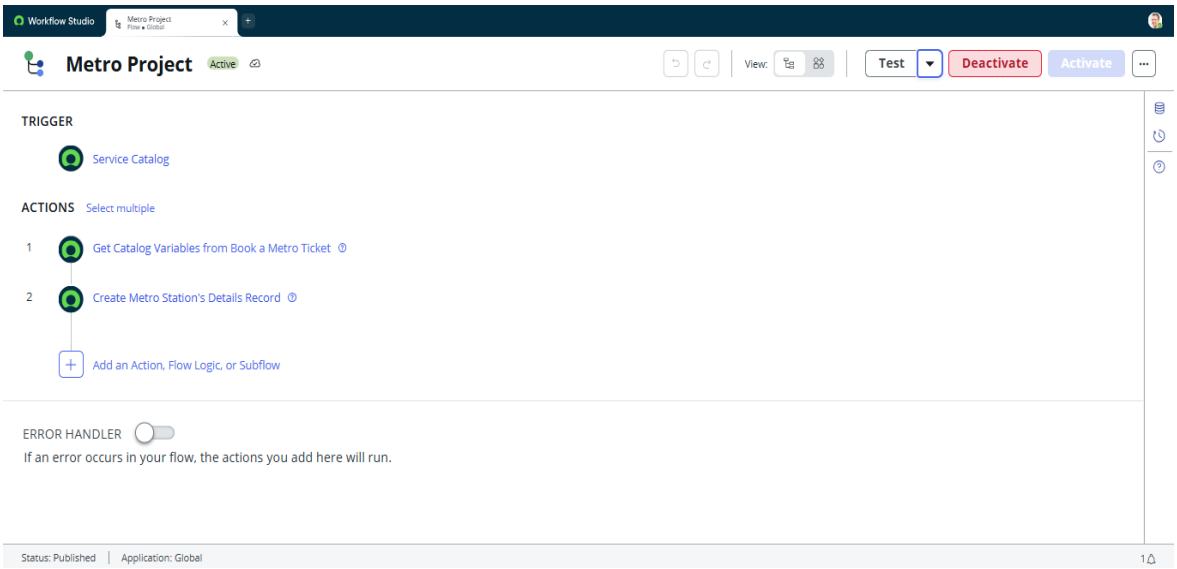
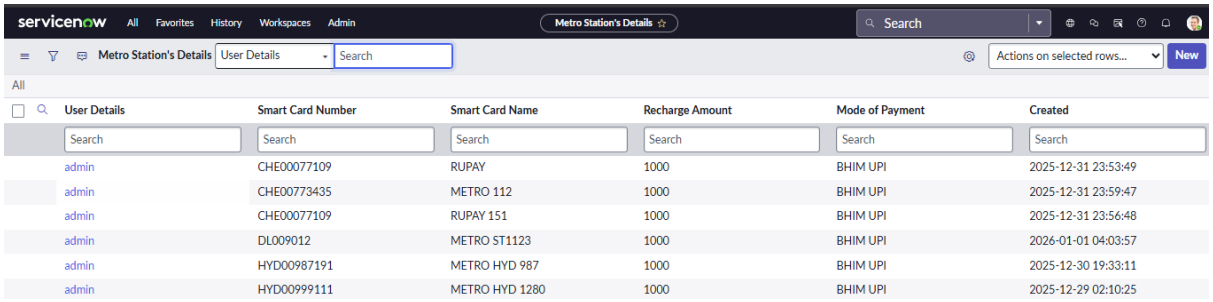


Figure 4: Flow Designer triggered automatically after Metro Ticket submission.

2. **Record Creation**: The system successfully created a structured record in the **Metro Database** (u_metro_station_s_details), ensuring data integrity.



User Details	Smart Card Number	Smart Card Name	Recharge Amount	Mode of Payment	Created
admin	CHE00077109	RUPAY	1000	BHIM UPI	2025-12-31 23:53:49
admin	CHE00773435	METRO 112	1000	BHIM UPI	2025-12-31 23:59:47
admin	CHE00077109	RUPAY 151	1000	BHIM UPI	2025-12-31 23:56:48
admin	DL009012	METRO ST1123	1000	BHIM UPI	2026-01-01 04:03:57
admin	HYD00987191	METRO HYD 987	1000	BHIM UPI	2025-12-30 19:33:11
admin	HYD00999111	METRO HYD 1280	1000	BHIM UPI	2025-12-29 02:10:25

Figure 5: Database record generated with mapped travel variables.

3. **QR Rendering**: Upon submission, the onSubmit Catalog Client Script executed successfully, fetching the sys_id and rendering a unique scannable QR code via a modal popup for the passenger.

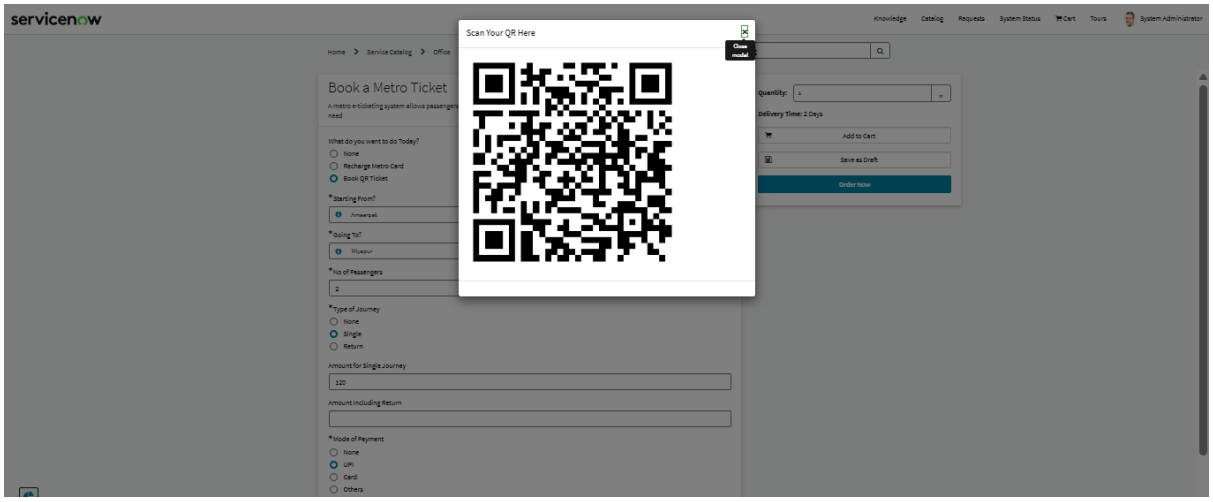


Figure 6: QR code ticket rendered for immediate traveler use.

4. **Flow Completion**: The execution details confirm that the backend lifecycle completed all actions from variable capture to record storage.

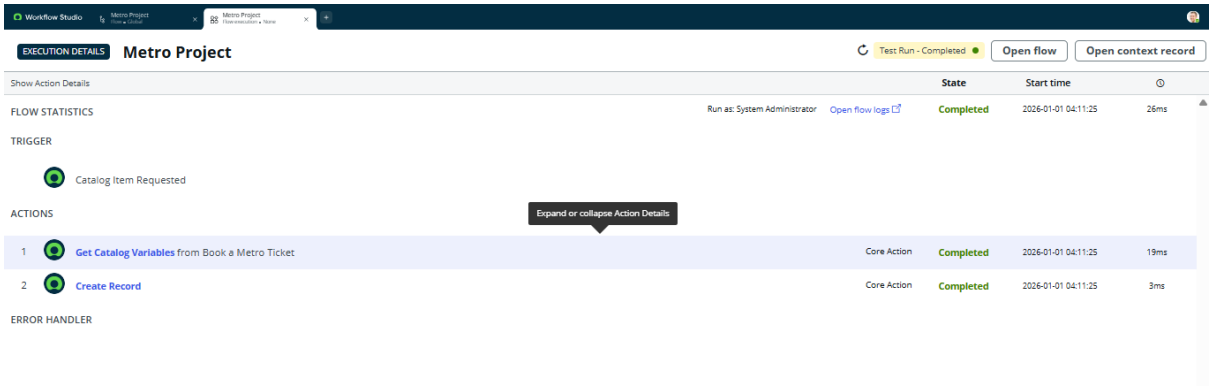


Figure 7: Flow execution completed successfully for Metro Project.

Security and Access Control (ACLs)

- **Role-Based Access**: Custom ACLs were verified to ensure that while commuters can book tickets or recharge cards, they cannot directly access or browse the backend database table.
- **Default Protections**: The four default ACLs created during table setup were utilized to maintain secure, role-restricted access data.

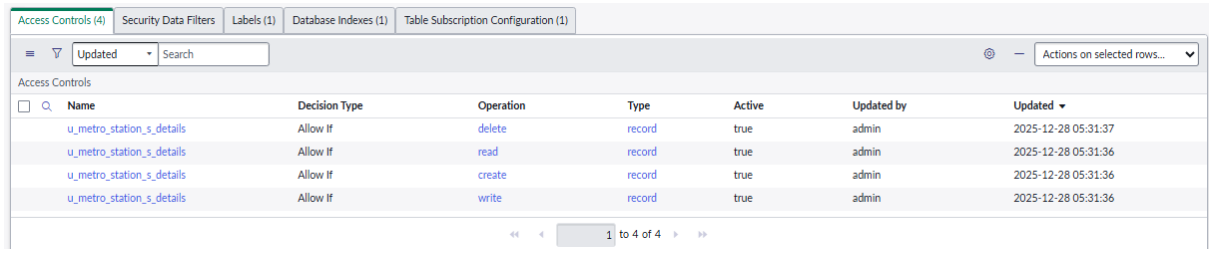


Figure 8: ACL Verification

Conclusion

The testing results confirm that the **Metro Ticket Generating System** meets all functional requirements. The solution performs reliably across all critical stages request submission, dynamic fare calculation, and instant QR rendering ensuring a stable and efficient deployment for urban transit.