

# FLOW DESIGNER

## 1. Introduction

This document describes the Flow Designer automation implemented for the Metro Ticket Booking and Smart Card Recharge system in ServiceNow. The flow ensures automatic data capture and record creation immediately after a user submits a metro-related request through the Service Catalog.

## 2. Flow Overview

Attribute	Description
Flow Name	Metro Project
Application Scope	Global
Flow Status	Active
Automation Type	Service Catalog-driven

The flow is designed to eliminate manual intervention by automatically processing metro ticket booking and recharge requests.

## 3. Trigger Configuration

### 3.1 Trigger Type

- Service Catalog Trigger

### 3.2 Trigger Condition

- Activated when a user submits the “Book a Metro Ticket” Service Catalog item

### 3.3 Trigger Purpose

- Ensures real-time automation
- Starts backend processing immediately after request submission

- Guarantees consistent handling of all metro transactions

## 4. Automation Workflow Design

The flow consists of sequential actions that capture user input and store it securely in the custom metro database.

## 5. Flow Actions Breakdown

### 5.1 Action 1 - Capture User Inputs

**Action Name:** Get Catalog Variables from Book a Metro Ticket Item

**Action Type:** Data Retrieval

**Objective:**

- Extract all values entered by the user in the Service Catalog form

**Captured Inputs:**

- Smart Card Number
- Smart Card Name
- Recharge Amount
- Mode of Payment
- User Details (Requested For)

This action acts as the foundation for accurate data mapping in subsequent steps.

---

### 5.2 Action 2 - Create Metro Transaction Record

**Action Name:** Create Metro Database Record  
**Action Type:** Create Record

**Target Table:**

- Metro Database (u\_metro\_database)

**Objective:**

- Store metro ticket booking and recharge information in a centralized data table

## 6. Field Mapping Strategy

### Metro Database Field Data Source

Smart Card Number	Catalog Variable - Smart Card Number
Smart Card Name	Catalog Variable - Smart Card Holder Name
Recharge Amount	Catalog Variable - Recharge Amount
Mode of Payment	Catalog Variable (UPI / Card / Net Banking)
User Details	Trigger → Requested For
Created	System-generated timestamp

This structured mapping ensures data accuracy, traceability, and audit compliance.

The screenshot shows the Workflow Studio interface for a project named "Metro Project". The workflow consists of two main steps:

1. Get Catalog Variables from Book A Metro Ticket Item
2. Create Metro Database Record

Under "Action Properties" for the second step, the "Action" is set to "Create Record". The "Action Inputs" section lists the following fields and their values:

- \* Table: Metro Database [u\_metro\_database]
- \* Fields:
  - Mode of Payment: UPI
  - Recharge Amount: 1 - Get Catalog → recharge\_amount
  - Smart Card Name: 1 - Get Catalog → smart\_card\_holder\_name
  - Smart Card Number: 1 - Get Catalog → smart\_card\_number
  - User Details: Trigger - Service → ... → Create...
  - Created: Trigger - Service → ... → Create...
- + Add field value

At the bottom, there is a "Add an Action, Flow Logic, or Subflow" button and an "ERROR HANDLER" section.

## **7. End-to-End Flow Execution Logic**

1. User submits the **Book a Metro Ticket** Service Catalog item
2. Service Catalog trigger activates the flow
3. Catalog variables are retrieved from the request
4. Flow Designer maps inputs to the u\_metro\_database table
5. A new metro transaction record is created automatically
6. Data is stored successfully for reporting and future reference

## **8. Automation Benefits**

- Eliminates manual data entry
- Ensures consistency across all metro transactions
- Supports accurate reporting and auditing
- Improves user experience through faster processing

## **9. Summary**

The Metro Ticket Booking Flow Designer automation provides a **reliable, scalable, and efficient backend workflow**. By integrating Service Catalog inputs with a custom metro database, the flow ensures seamless transaction handling, data integrity, and compliance with ServiceNow automation best practices.