

# **Metro Ticket Generating System**

## **Data Architecture**

### **Introduction**

The purpose of this document is to describe the data model and table structure used in the Metro Ticket Generating System developed in ServiceNow. The data architecture ensures that all transit-related requests, station details, and fare transactions are stored in a structured, secure, and auditable format to support automation, reporting, and tracking. By defining the policies and rules that govern data flow, this architecture supports the primary business objective of digitizing metro operations.

### **Overview of Custom Table**

To store and manage metro ticketing information, a custom table has been created to act as the central data repository for all service requests submitted through the Service Catalog.

<b>Attribute</b>	<b>Value</b>
<b>Table Label</b>	Metro Station's Details
<b>Table Name</b>	u_metro_station_s_details
<b>Application Scope</b>	Global
<b>Purpose</b>	Store structured data captured from Service Catalog metro ticket requests.

### **Custom Table: u\_metro\_station\_s\_details**

The u\_metro\_station\_s\_details table stores all essential details related to the passenger's journey, including station references, passenger count, and work status. This table is populated automatically through Flow Designer automation immediately upon the submission.

Table Metro Station's Details						
<div>Columns Controls Application Access</div> <div>Table Columns Reference Search</div> <div>1 to 12 of 12</div> <div>New</div>						
Dictionary Entries	Column label	Type	Reference	Max length	Default value	Display
	Updated by	String	(empty)	40	false	false
	Sys ID	Sys ID (GUID)	(empty)	32	false	false
X	Mode of Payment	String	(empty)	40	false	false
	Created by	String	(empty)	40	false	false
	Updated	Date/Time	(empty)	40	false	false
X	Smart Card Name	String	(empty)	40	false	false
X	Station Name	String	(empty)	40	false	false
	Created	Date/Time	(empty)	40	false	false
X	User Details	String	(empty)	40	false	false
X	Smart Card Number	String	(empty)	40	false	false
	Updates	Integer	(empty)	40	false	false
X	Recharge Amount	String	(empty)	40	false	false
	Insert a new row...					

Figure 1:Table Metro Station’s Details

Column Label	Type	Max Length	Display
Updated by	String	40	false
Sys ID	Sys ID (GUID)	32	false
Mode of Payment	String	40	false
Created by	String	40	false
Updated	Date/Time	40	false
Smart Card Name	String	40	false

<b>Column Label</b>	<b>Type</b>	<b>Max Length</b>	<b>Display</b>
<b>Station Name</b>	String	40	false
<b>Created</b>	Date/Time	40	false
<b>User Details</b>	String	40	false
<b>SmartCard Number</b>	String	40	false
<b>Updates</b>	Integer	40	false
<b>Recharge Amount</b>	String	40	false

## Field Properties

To ensure data integrity and process flow efficiency, specific field properties are defined:

- **Reference Fields:** Fields such as Source and Destination reference the Metro Station Details table to leverage platform capabilities and ensure data consistency.

- **Choice Fields:** The Work Status is configured as a choice field to maintain standardized request status values (e.g., Active, Expired).
- **System Fields:** Fields such as Sys ID, Created, and Updated are system-generated and read-only, providing a permanent audit trail.
- **Mandatory Fields:** Key fields like Ticket ID and Work Status are enforced to ensure complete and compliant data entry.

## Table Relationships

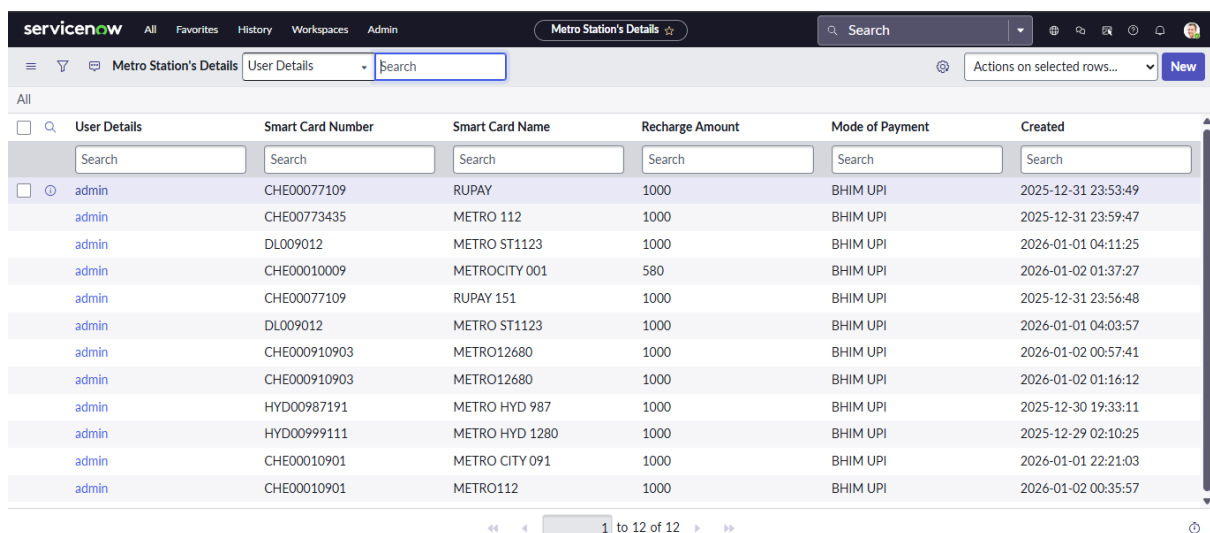
The `u_metro_station_s_details` table maintains relationships with existing ServiceNow tables to leverage platform capabilities:

- **User Table (`sys_user`):** Used for passenger and requester references to ensure accountability.
- **Metro Station Details:** Used to route requests based on specific geographic locations and fare zones.
- **Access Control:** These relationships ensure role-based access control and seamless integration with ServiceNow task management.

# Data Flow Overview

The data lifecycle within the application follows this structured flow:

1. **Submission:** The user submits a metro ticket request via the Service Catalog.
2. **Capture:** Catalog variables are captured using Flow Designer.
3. **Storage:** Data is mapped and stored in the `u_metro_station_s_details` custom table.



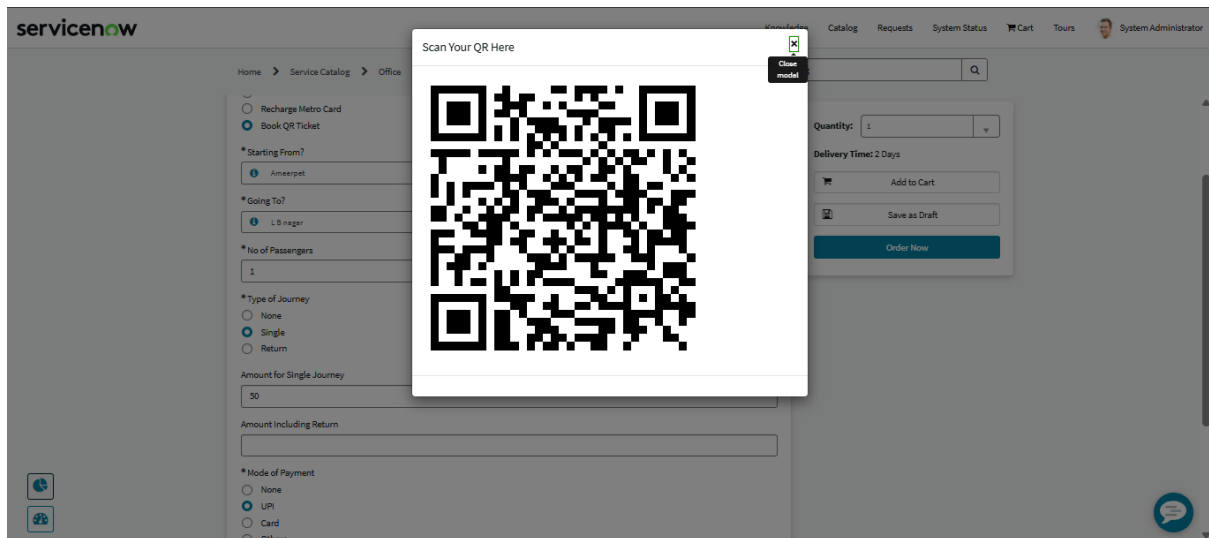
The screenshot displays the ServiceNow interface for the 'Metro Station's Details' table. The table is filtered by 'User Details' and shows 12 records. The columns are: User Details, Smart Card Number, Smart Card Name, Recharge Amount, Mode of Payment, and Created. The records are sorted by Created date in descending order.

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	DL009012	METRO ST1123	1000	BHIM UPI	2026-01-01 04:11:25
admin	CHE00010009	METROCITY 001	580	BHIM UPI	2026-01-02 01:37:27
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	CHE000910903	METRO12680	1000	BHIM UPI	2026-01-02 00:57:41
admin	CHE000910903	METRO12680	1000	BHIM UPI	2026-01-02 01:16:12
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
admin	CHE00010901	METRO CITY 091	1000	BHIM UPI	2026-01-01 22:21:03
admin	CHE00010901	METRO112	1000	BHIM UPI	2026-01-02 00:35:57

**Figure 1:** Proof of Data Storage — Records successfully populated in the `u_metro_station_s_details` table.

s

4. **Generation:** A unique QR code is generated using a Catalog Client Script for immediate commuter use.



**Figure 2:** Data Generation Outcome A unique QR code rendered instantly via spModal upon request submission.

**5. Tracking:** Records are updated automatically as the journey progresses, with the final status recorded for reporting purposes.

## Conclusion

The data architecture of the Metro Ticket Generating System ensures a clean, scalable, and auditable data structure. By using a custom table integrated with ServiceNow's native user and automation capabilities, the system supports efficient fulfilment, reliable tracking, and compliance with modern transit management best practices.