

# **Metro Ticket Generating System**

## **User Flow**

### **Introduction**

This document describes the User Interface (UI), User Experience (UX), and navigation flow of the **Metro Ticket Generating System** developed using ServiceNow. The primary objective is to ensure an intuitive, efficient, and user-friendly process for submitting transit requests and generating digital tickets.

### **Service Portal Procedure**

To access the system and generate a ticket, follow this step-by-step manual for the Service Portal:

- 1. Login to ServiceNow PDI:** Access your Personal Developer Instance using admin credentials.
- 2. Access the Portal:** Copy the instance domain and add the prefix /sp to the URL (e.g., <https://devXXXXX.service-now.com/sp>).
- 3. Search for Catalog Item:** Use the search bar to locate the **Book A Metro Ticket** item.

4. **Fill and Submit:** Enter travel details, including stations and passenger count, then click **Submit**.
5. **View Request:** A new request number is generated, and ticket details are displayed on the **Requested Item Page**.

## UI & UX Considerations

The **Book a Metro Ticket** catalog item is designed with a clean and structured form layout to improve readability and reduce user effort.

- **Clearly Labeled Fields:** Ensures commuters understand required inputs like "Starting From" and "Going To".
- **Mandatory Indicators:** Visually identifies required travel information to prevent submission errors.
- **Real-time Fare Preview:** Provides an automated cost calculation before the user commits to the submission.
- **Dynamic UI Behaviour:** UI Policies and Client Scripts manage field visibility so that fields appear or hide based on the specific request type selected (e.g., hiding Smart Card fields during QR booking).

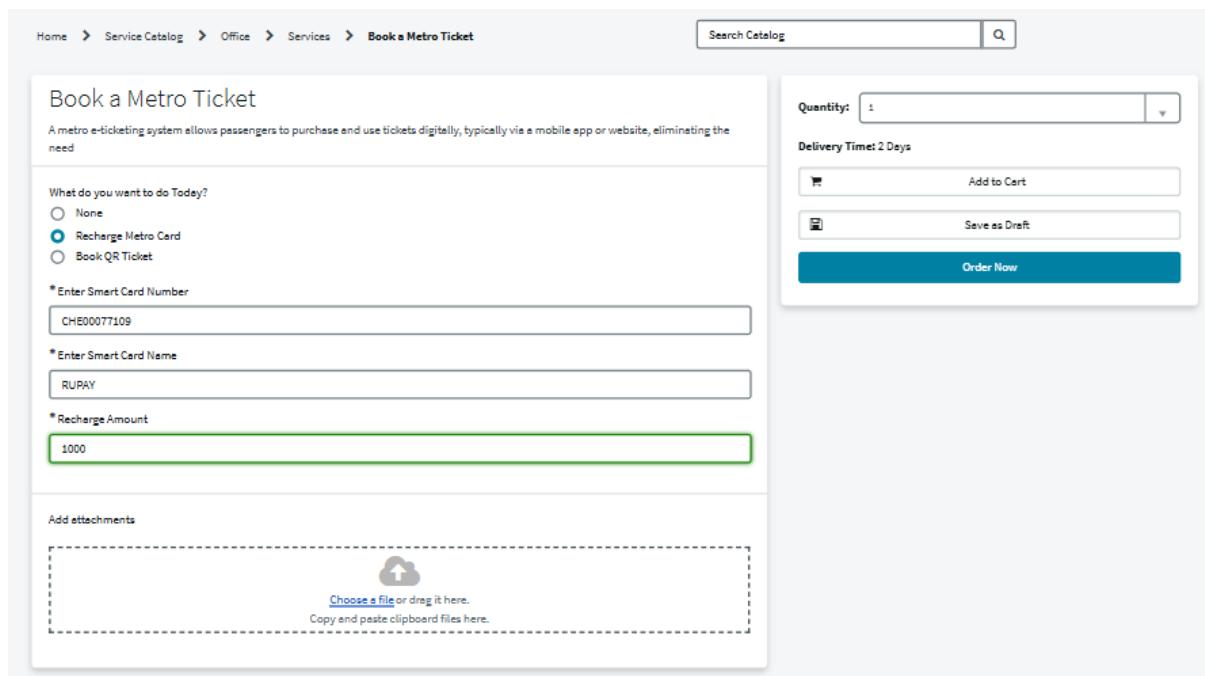
## Dynamic UI Behaviour (User Path Logic)

To improve the User Experience (UX), UI Policies and Client Scripts manage field visibility so that fields appear or hide based on the specific selection made in the "**What do you want to do Today?**" variable.

### Option A: Smart Card Recharge Flow

When a user selects **Recharge Metro Card**, the form dynamically adapts to show card-specific fields while hiding journey variables.

- **Fields Displayed:** Smart Card Number, Smart Card Name, and Recharge Amount.
- **Table Mapping:** Data captured here is mapped to the **Metro Station's Details** (u\_metro\_station\_s\_details) table.



The screenshot shows a web-based booking interface for a "Book a Metro Ticket". The top navigation bar includes links for Home, Service Catalog, Office, Services, and Book a Metro Ticket. A search bar is also present. The main content area is titled "Book a Metro Ticket" and contains the following fields:

- A descriptive note: "A metro e-ticketing system allows passengers to purchase and use tickets digitally, typically via a mobile app or website, eliminating the need for physical tickets."
- A radio button group for "What do you want to do Today?":
  - None (unchecked)
  - Recharge Metro Card (checked)
  - Book QR Ticket (unchecked)
- Input fields for "Enter Smart Card Number" (CHE00077109), "Enter Smart Card Name" (RUPAY), and "Recharge Amount" (1000).
- A section for "Add attachments" with a placeholder for file uploads.
- A sidebar on the right side of the form:
  - Quantity: 1
  - Delivery Time: 2 Days
  - Action buttons: Add to Cart, Save as Draft, and a prominent Order Now button.

**Figure 1:** Smart Card Recharge Interface Dynamic UI displays card variables and hides travel-specific fields.

The screenshot shows a user interface for a 'Smart Card Recharge' task. At the top, there's a navigation bar with 'Home > My Request - RITM0010031'. Below it, a summary box displays the request number 'RITM0010031', creation and update times ('19m ago'), and status ('Open'). A descriptive text box explains that a metro e-ticketing system allows passengers to purchase and use tickets digitally via a mobile app or website. The main form area has tabs for 'Activity', 'Attachments', and 'Additional Details', with 'Additional Details' being active. It contains fields for 'Smart Card Number' (DL009012), 'Smart Card Name' (METRO ST1123), 'Recharge Amount' (1000), and 'Mode of Payment' (None). A note at the top says 'What do you want to do Today? Recharge Metro Card'.

**Figure 2:** Dynamic UI for Smart Card Recharge — Displays card-specific variables like Smart Card Number and Recharge Amount while hiding journey fields.

## Key Technical Details:

- **Field Mapping:** These variables map directly to the **Metro Station's Details** (u\_metro\_station\_s\_details) table.
- **Table Schema:** The backend table includes fields for "Smart Card Name," "Smart Card Number," and "Recharge Amount".

## Option B: Book QR Ticket Flow

When a user selects **Book QR Ticket**, the form reveals journey-specific variables and calculates costs in real-time.

- **Fields Displayed:** Starting From, Going To, No of Passengers, Type of Journey, and Mode of Payment.
- **Fare Preview:** Displays the "Amount for Single Journey" or "Amount Including Return" based on the selected stations.

The screenshot shows a web-based ticket booking interface. At the top, there's a navigation bar with links: Home > Service Catalog > Office > Services > Book a Metro Ticket. To the right of the navigation is a search bar labeled 'Search Catalog' with a magnifying glass icon. Below the navigation, the main title is 'Book a Metro Ticket'. A descriptive subtitle explains: 'A metro e-ticketing system allows passengers to purchase and use tickets digitally, typically via a mobile app or website, eliminating the need for physical tickets.' On the left side of the form, there are several input fields and dropdown menus:

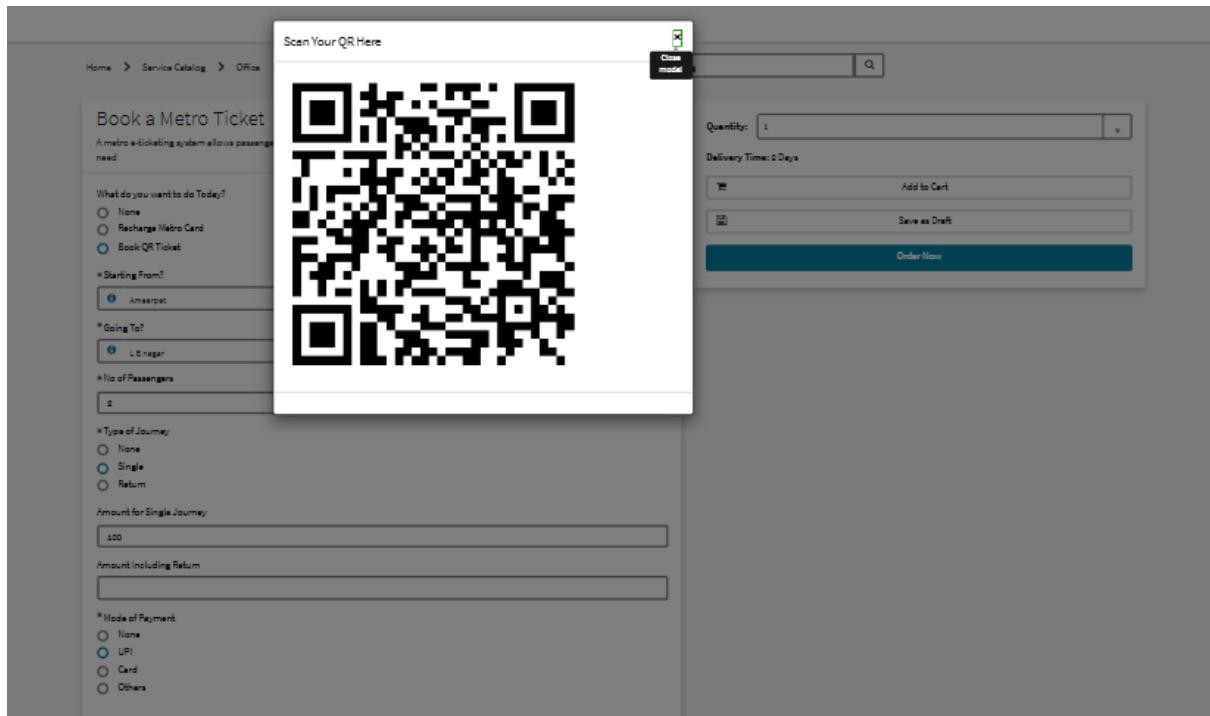
- 'What do you want to do Today?' with options: None (radio), Recharge Metro Card (radio), and Book QR Ticket (radio, selected).
- 'Starting From?' dropdown menu showing 'Amarapet' (selected).
- 'Going To?' dropdown menu showing 'Lalnagar' (selected).
- 'No of Passengers' input field showing '2'.
- 'Type of Journey' with options: None (radio), Single (radio, selected), and Return (radio).
- 'Amount for Single Journey' input field showing '100'.
- 'Amount Including Return' input field (empty).
- 'Mode of Payment' with options: None (radio), UPI (radio, selected), Card (radio), and Others (radio).

On the right side of the form, there are delivery options:

- 'Quantity:' dropdown menu showing '1'.
- 'Delivery Time: 2 Days'
- 'Add to Cart' button (disabled, greyed out).
- 'Save as Draft' button.
- A large blue 'Order Now' button.

At the bottom left, there's a section for 'Add attachments' with a dashed box and a cloud icon, followed by the text 'Choose a file or drag it here. Copy and paste clipboard files here.'

**Figure 3:** QR Ticket Booking Interface — System reveals station selection and passenger count fields.



**Figure 4:** QR Ticket Booking Interface — System shows QR code for payment .

## Key Technical Details:

- **Trigger Selection:** The user selects "Book QR Ticket" from the primary choice variable.
- **Real-time Fare Preview:** The "Amount for Single Journey" field auto-populates (e.g., 100) based on the stations selected, such as "Ameerpet" to "L B Nagar".
- **Mandatory Fields:** Journey details and "Mode of Payment" become mandatory to ensure complete transaction data.

## **QR Mapping and Fulfilment Process**

The fulfilment flow is fully automated to ensure transparency and accountability.

- **QR Mapping:** Upon submission, the system executes an onSubmit Catalog Client Script that maps the unique record sys\_id to a QR code API.
- **Instant Rendering:** The QR code is displayed immediately in an spModal popup, allowing the user to scan and enter the metro station without manual staff intervention.
- **Data Tracking:** The request variables are mapped from the catalog item to the custom **Metro Database** (u\_metro\_station\_s\_database) for permanent record storage.

## **Validation and Benefits**

The UI/UX design was validated based on ease of navigation and error prevention through automated validations.

- **Improved Satisfaction:** Commuters benefit from faster request fulfilment.
- **Reduced Errors:** Dynamic fields and mandatory indicators minimize incorrect submissions.
- **Standardized Delivery:** Every ticket follows a predictable, automated logic path.

## **Conclusion**

The combined UI, UX, and navigation design of the **Metro Ticket Generating System** ensures a smooth and efficient experience for end users. The implementation of dynamic forms, real-time fare previews, and automated QR generation significantly improves usability and operational efficiency in a high-traffic transit environment.