

Metro Ticket Generating System

Deployment, Documentation & Final Presentation

Introduction

This final phase marks the transition of the **Metro Ticket Generating System** from a development environment to a stakeholder-ready deployment. Phase 5 focuses on rigorous troubleshooting, the creation of a comprehensive technical blueprint, and the preparation of a visual demonstration. The goal is to ensure the system is not only functional but also maintainable, scalable, and clearly documented for future administrators and developers.

Troubleshooting & Debugging

Before final deployment, the system underwent deep technical validation to ensure reliability.

- **Logic Debugging:** Debugged QR encoding issues and potential fare mismatches using logs and print statements within the scripts.
- **Flow Execution Logs:** Traced every step of the **Metro Project** flow using Flow Designer's **Execution Details** to verify that condition paths for record creation triggered correctly.

- **Dynamic UI Validation:** Verified that selecting the "Recharge Metro Card" option correctly hides journey-specific fields and reveals smart card input variables, ensuring a focused user experience and data accuracy.

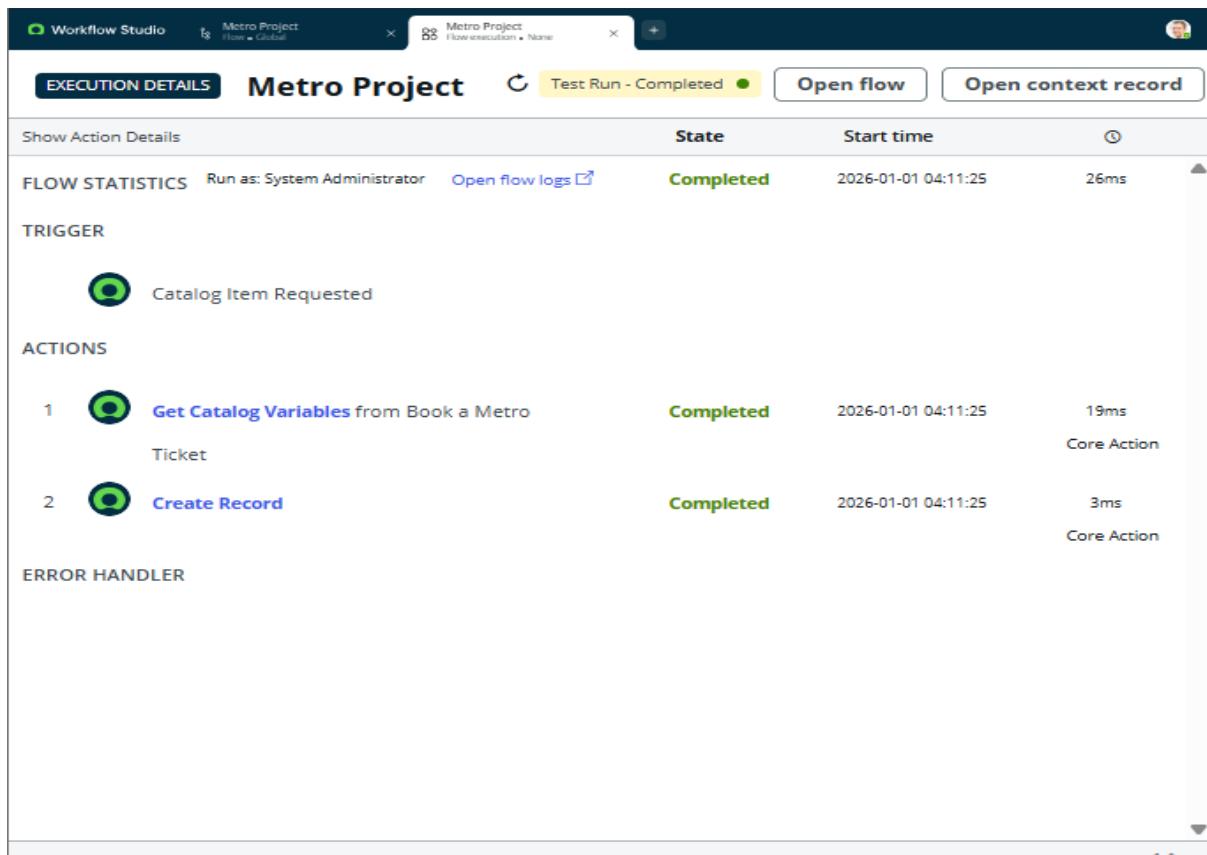


Figure 1: A screenshot of Flow Designer Execution Details showing a "Completed" path.

- **Variable Alignment:** Resolved data binding issues using **Catalog Client Scripts** to ensure that station selections dynamically populated the correct fare values on the portal.

- **Form Behaviour Testing:** Confirmed that UI policies correctly managed field visibility for both "Recharge" and "Booking" paths via Service Portal preview mode.

The screenshot shows the ServiceNow Service Catalog interface. The top navigation bar includes links for Knowledge, Catalog, Requests, System Status, Cart, Tours, and System Administrator. The main page title is 'Book a Metro Ticket'. Below the title, a sub-header states: 'A metro e-ticketing system allows passengers to purchase and use tickets digitally, typically via a mobile app or website, eliminating the need for paper tickets.' A radio button group labeled 'What do you want to do Today?' has 'Recharge Metro Card' selected. An 'Add attachments' section contains a dashed box for file uploads. To the right, a sidebar displays 'Quantity: 1' and 'Delivery Time: 2 Days'. It includes three buttons: 'Add to Cart', 'Save as Draft', and a large blue 'Order Now' button.

Figure 2a: Default State (None) – Shows the initial "clean" form layout where journey and card variables are hidden to minimize user effort

This screenshot shows the same ServiceNow Service Catalog interface as Figure 2a, but with a different selection. The 'Recharge Metro Card' radio button is now selected. This triggers the dynamic reveal of three input fields: '* Enter Smart Card Number' containing 'CH000900012', '* Enter Smart Card Name' containing 'METRO190', and '* Recharge Amount' containing '1000'. The rest of the form and sidebar remain identical to Figure 2a.

Figure 2b: Recharge Metro Card Selection – Demonstrates the dynamic reveal of Smart Card variables (Number, Name, and Amount) based on user selection.

The screenshot shows the ServiceNow interface for booking a QR ticket. The main form fields include:

- Starting From:** Ameerpet
- Going To:** LB Nagar
- No of Passengers:** 5
- Type of Journey:** Single (selected)
- Amount for Single Journey:** 250
- Mode of Payment:** UPI (selected)

A sidebar on the right displays delivery information and action buttons:

- Quantity: 1
- Delivery Time: 2 Days
- Action Buttons: Add to Cart, Save as Draft, Order Now

Figure 2c: Book QR Ticket Selection – Displays the journey-specific variables, including Source/Destination stations and the **Real-time Fare Preview**.

Adherence to Timelines

The project followed a disciplined **Sprint-based approach** to meet all 2025 graduation milestones:

- **Milestone 1:** Catalog Creation.
- **Milestone 2:** Form Setup & Dynamic Logic.
- **Milestone 3:** Testing, Security (ACLs), and QR Integration.
- **Milestone 4:** Deployment & Documentation.

Innovation & Maintenance

The system was built with a "Low-Code First" mindset to ensure it is easy to hand off to future IT teams:

- **No-Code Automation:** Leveraged **Flow Designer** for the entire backend lifecycle, eliminating the need for complex custom Script Includes or Business Rules.
- **Efficient Scripting:** Kept Client Scripts and UI Policies to the minimal essential use required for dynamic front-end behaviour.

Technical & Functional Blueprint

A complete repository of the system's architecture was prepared to support replicability:

- **Functional Overview:** Clear definitions of form variables, user experience flows, and the QR mapping process.
- **QR Generation Snippet:** A Client Script uses the spModal API to render a QR code using an external encoding service, passing the `g_form.getUniqueValue()` as the data payload.
- **Flow Designer Logic:** The "Metro Project" flow is triggered by a **Service Catalog** request. It executes two core actions: **Get Catalog Variables** to retrieve user inputs and **Create Record** to log the transaction in the backend database.

- **Technical Blueprint:** Included QR generation code snippets, Flow Designer logic snapshots, and the **Metro DataBase** custom table schema.
- **Setup Manual:** A step-by-step guide for recreating the **Catalog → Flow → ACL** structure in any new Personal Developer Instance (PDI).

Visual Demonstration & Project Demo

To present the system to stakeholders, a multi-role demonstration was planned:

- **Portal Walkthrough:** A guided tour from the Portal Home to the **Book A Metro Services** category.

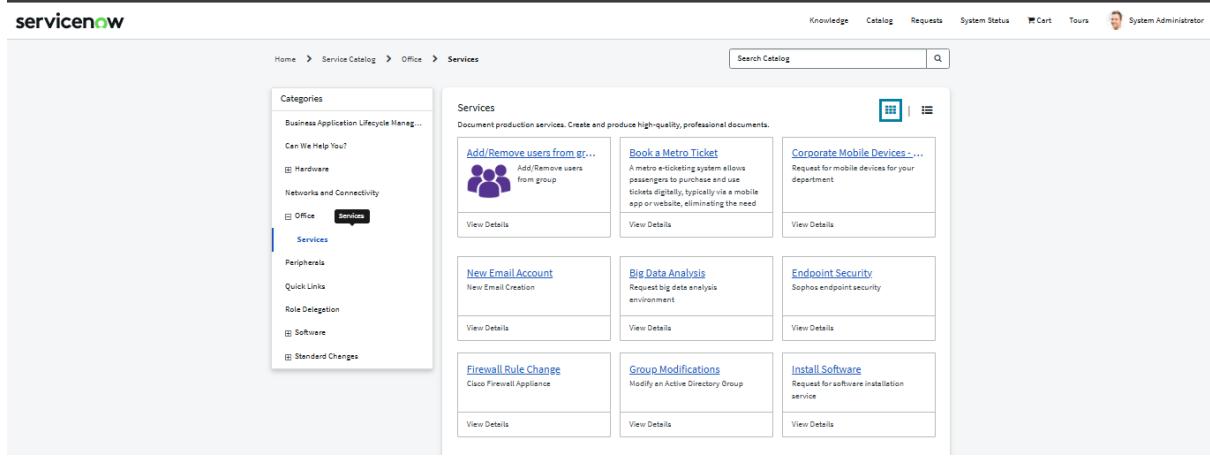
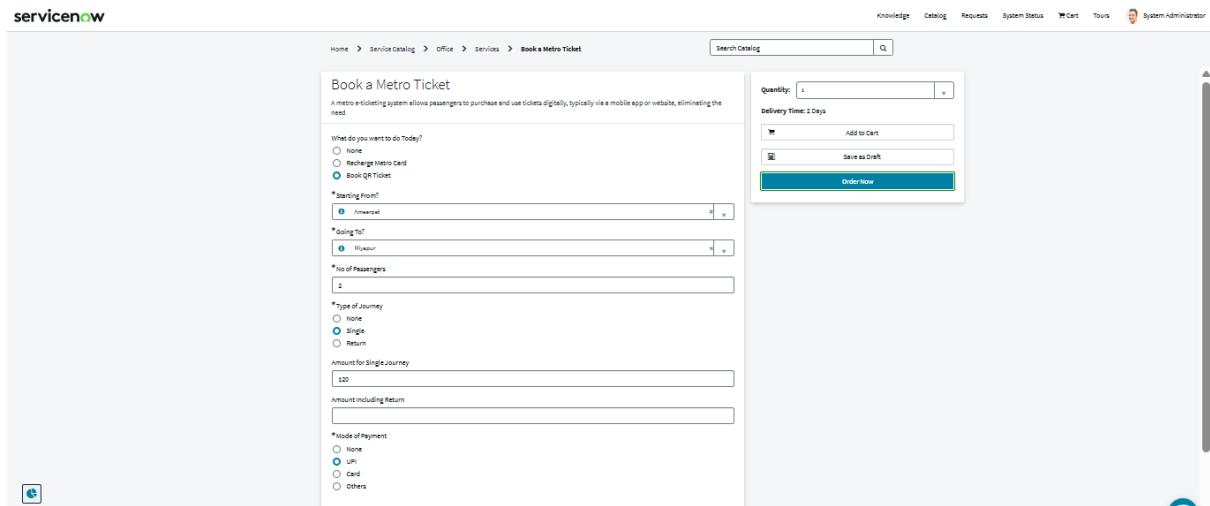


Figure 3: Service Catalog portal in service category

- **Live Submission:** A real-time demonstration of a request being submitted and the QR code rendering instantly via spModal.



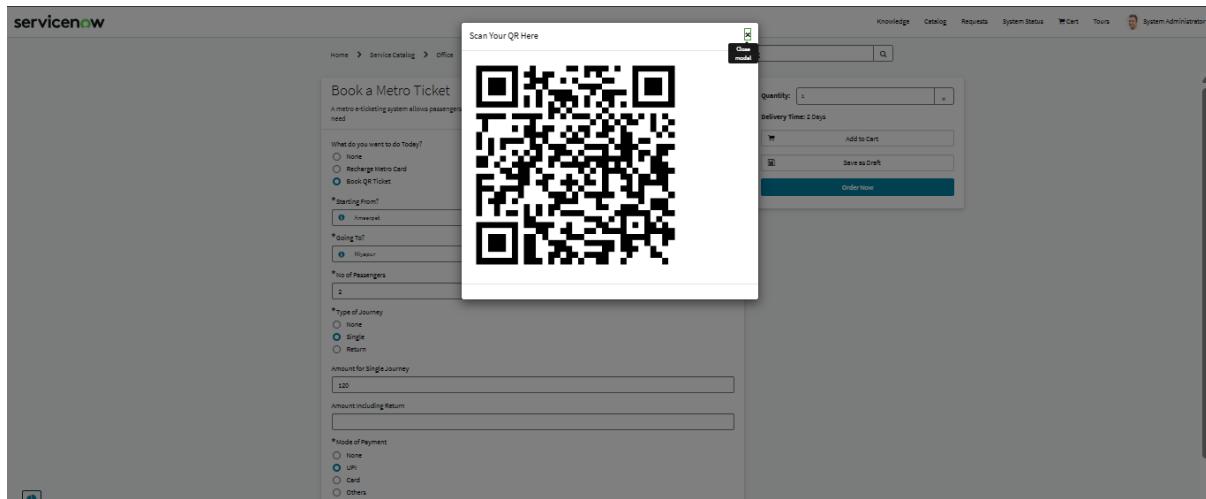


Figure 4: A screenshot of the **Service Portal** form showing the "Order Now" button and the resulting **QR Code**.

- **Data Storing:** Proof of the data being successfully logged in the backend **Metro Station's Details** table.

Metro Station's Details					
	User Details	Smart Card Number	Smart Card Name	Recharge Amount	Mode of Payment
	Search	Search	Search	Search	Search
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	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: A screenshot of the Metro Station's Details table list view showing stored transaction data.

- **Annotated Visuals:** High-quality screenshots of the Catalog UI, QR output, and Flow logs were annotated to explain each stage clearly.

Scalability & Future Roadmap

The system is designed to grow beyond its current functional scope:

- **City-Wide Integration:** Reflects actual city metro workflows with dynamic fare bands.
- **Future Channels:** Roadmap includes implementing **SMS and Email-based ticketing** for commuters without portal access.
- **Advanced Features:** Plans for payment gateway integration and a **Travel Analytics Dashboard** for station authorities.

Conclusion

The Phase 5 successfully transitions the **Metro Ticket Generating System** from development to a fully documented, deployable solution. By combining robust troubleshooting, clear technical blueprints, and a roadmap for future scalability, the system ensures a high-quality transit experience aligned with ServiceNow best practices.