Problem Specification:

NewYork Metro station wants to establish a TicketDistributor machine that issues tickets for passengers travelling in metro rails. Travelers have options of selecting a ticket for a single trip, round trips or for multiple trips. They can also issue a metro pass for regular passengers or a time card for a day, a week or a month according to their requirements. The discounts on tickets will be provided to frequent travelling passengers. The machine is also supposed to read the metro pass and time cards issued by the metro counters or machine. The ticket rates differ based on whether the traveler is a child or an adult. The machine is also required to recognize original as well as fake currency notes. The typical transaction consists of a user using the display interface to select the type and quantity of tickets and then choosing a payment method of either cash, credit/debit card or smartcard. The ticket or tickets are printed and dispensed to the user. Also the messaging facilities after every transaction are required on the registered number. The system can also be operated comfortably by a touch-screen. A large number of heavy components are to be used. We do not want our system to slow down, and also usability of the machine. The TicketDistributor must be able to handle several exceptions, such as aborting the transaction for incomplete transactions, insufficient amount given by the travelers to the machine, money return in case of aborted transaction, change return after successful transaction, showing insufficient balance in the card, updated information printed on the tickets e.g. departure time, date, time, price, valid from, valid till, validity duration, ticket issued from and destination station. In case of exceptions, an error message is to be displayed. We do not want user feedback after every development stage but after every two stages to save time. The machine is required to work in a heavy load environment such that at the morning and evening time in weekdays, and in weekends performance and efficiency would not get affected.

Questions:

Q-1. Considering the given problem description of the NewYork TicketDistributor system, identify the stakeholders and end-users of the system. What kind of elicitation techniques will you apply to gather requirements from different stakeholders? Justify why and how the chosen elicitation techniques are helpful in gathering the requirements?

Answer :

Considering the given problem description,
The stakeholders are :

1. **Metro authorities** that set the ticket fares, discounts, schedules.
2. **Government entities** that oversee the compliance of fare policies and ensure that railway rules and regulations are followed.
3. **Developers** that design and implement the software and hardware of the system.
4. **Security teams** that identify fraudulent activity (for eg., cases of fake notes or fake smart cards).
5. **Banking and payment gateway providers** who handle payments.
6. **Finance department** which handles revenue from ticket sales, and manages refunds.

The end users are :

- **Travelers**

Some elicitation techniques I would apply to gather requirements from different stakeholders are :

1. **Interviews**

Why → Interviews with stakeholders help to gain insight on the functional and non-functional requirements

How → Structured interviews could be conducted for metro authorities and developers to understand operational needs and semi-structured interviews with passengers to gain insight into how to enhance user experience.

2. **Questionnaires**

Why → We can ensure a large amount of input

How → Circulate online forms and at metro stations to collect feedback on their preferences and to gain information on potential pain points.

3. **Domain Analysis**
   Why → In domain analysis we examine existing and related documentation and applications which is a useful way to help us understand the existing metro ticketing systems and ensures that they are compatible with the current infrastructure
   How → Study documentation and policies from similar metro systems

4. **Task analysis**
   Why → Using task analysis, we can observe each step that a passenger takes in the ticket purchasing process
   How → Decomposition of ticket selection and payment process into smaller tasks

5. **Observation**
   Why → Observation helps identify issues that users face while interacting with existing ticketing systems.
   How → Observing users at metro stations

6. **Prototyping**
   Why → Prototyping allows stakeholders to gather relevant feedback and work on their requirements accordingly
   How → Developing a prototype to test interface with users

7. **Scenarios**
   Why → To understand and validate requirements and for test case development because they help to cover different ticketing scenarios
   How → By developing detailed use case diagrams and stories

8. **Joint Application Development (JAD) sessions**
   Why → Through these sessions, all stakeholders can be involved in the discussion to solve problems and therefore solutions can be attained quickly according to the needs and desires of the users
   How → By holding well structured sessions with the stakeholders with clear allotment of roles and responsibilities

Q-2. Enlist all functionalities of the TicketDistributor system by each user of the system, in the form of user stories (both front and back of the card).

a. Prioritize them (using the requirement prioritization techniques, e.g., AHP, Numerical Assessment, MoSCoW method, etc.), keeping priorities of non-functional aspects into consideration? How? Provide details.

Sr. No.	**Front of the Card**	**Back of the card**
1.	As a **traveler**, I want to **select a ticket type** so that **I can travel as per my requirement.**	Success - Ticket selected and added to purchase Failure - Display error message (a) "Ticket selected is invalid. Select a valid option" (b) "System error. Please try again."
2.	As a **traveler**, I want to **have multiple options for payment** so that I can **have flexibility while making payments.**	Success - Purchase has been paid for successfully and ticket(s) have been issued. (a) User selects a payment method and concludes the payment. (b) User uses a smart card, and the balance is updated (deducted) accordingly. Failure - Display error message (a) "Invalid card details. Please try again." (b) "Insufficient balance in smart card. Please recharge or try other payment options." (c) "Insufficient amount paid. Refund initiated." (d) "Payment gateway error. Please try again in some time or choose other

		payment options.” (e) “System error. Please try again.” (f) “Unable to dispense ticket. Refund initiated.”
3.	As a **traveler**, I want **correct change returns and refunds** so that **I don’t lose money**.	Success - Correct change/refund is received (a) User makes payment and system calculates and returns change accordingly. Failure - Display error message (a) “Unable to return change. Contact customer service.”
4.	As a **traveler**, I want **the machine to detect fake notes** so that **fraudulent transactions do not occur.**	Success - Original notes are accepted. (a) System recognizes and authenticates original notes and proceeds with the transaction. Failure - Displays error message (a) “Fake notes detected. Transaction aborted.” (b) “Currency not recognized. Please use a valid note.”
5.	As a **traveler**, I want to **issue a metro pass or a time card** so that I can **travel as per my requirements easily**.	Success - Metro pass or time card is successfully issued (a) User selects the type of pass (daily, weekly, monthly). Failure - Display error message (a) “Invalid option selected. Please select a valid option.” (b) “Insufficient balance on smart card.” (c) “System error. Please try again later.” (d) “User already has an active pass.”
6.	As a **frequently travelling**	Success - Discount is successfully

	passenger, I want **discounts on my tickets** so that **I can save money.**	applied. (a) User completes the payment at the discounted price. Failure - Display error message (a) “Discount not applicable. Minimum travel criteria not met.” (b) “Payment gateway error. Please try again later.” (c) “System error. Try again later.”
7.	As a **traveler,** I want **the machine to read my metro pass or time cards and update my remaining trips accordingly** so that I can **use them effectively.**	Success - Card is recognized and remaining trips are updated accordingly Failure - Display error message (a) “Card expired. Please issue a new one.” (b) “Zero remaining trips. Please issue a new pass.” (c) “Card not recognized. Please try again.” (d) “System error. Please try again later.”
8.	As a **traveler**, I want to **receive notifications about the successful completion of my transaction** so that **I know that I have successfully issued my tickets.**	Success - SMS notification is sent Failure - Display error message (a) “SMS delivery failed. Please check the registered number.” (b) “System error. Unable to send confirmation.”
9.	As a **traveler**, I want **refunds in case of aborted transactions**, so that **I don’t lose money**.	Success - Money is successfully refunded Failure - Display error message (a) “Unable to initiate a refund. Contact customer service.”
10.	As a **metro authority**, I want to **set ticket prices and discounts** so that **prices can be adjusted as needed and revenue**	Success - Ticket prices and discounts are updated. Failure - Display error message (a) “Error updating prices.

	targets are met.	Please try again.” (b) “Invalid price/discount update. Enter a valid option or change settings.”
11.	As a **security team**, I want the **system to record suspicious transactions** so that **we can investigate fraud cases.**	Success - Suspicious transactions are recorded. (a) Someone tries multiple failed login attempts. Failure - Display error message (a) “System error”
12.	As a **payment provider**, I want **payments to be encrypted** so that **transactions and related information remain secure.**	Success - Transactions are encrypted and secure. Failure - Display error message (a) “Encryption error. Transaction aborted.”
13.	As a **finance officer**, I want to **track revenues from sales** so that **accurate financial records are maintained**	Success - Revenue is recorded Failure - Display error message (a) “System error. Please try again later.”
14.	As a **developer**, I want **the machine to work in a heavy load environment** so that **it can function efficiently during peak hours.**	Success - The machine works efficiently Failure - Display error message (a) “System is taking too long to respond. Please wait or try again.” (b) “System error. Please try again later.”

Prioritizing them using the MoSCoW Technique :

User Story	Priority	Details
1. Selecting ticket type	**M**	It is a core function.
2. Multiple payment options	**M**	Ensures flexibility for users
3. Correct change returns	**M**	Prevents financial loss

4. Fake note detection	**M**	Prevents fraudulent activities
5. Issuing metro pass/ time card	**M**	Essential for frequent travelers
6. Discount for frequent travelers	**C**	A desirable requirement for frequent travelers but not a necessary one
7. Reading metro pass/time card	**M**	Necessary for validity of passes
8. SMS notification	**S**	Useful for tracking transactions
9. Refund	**S**	To prevent loss of money
10. Setting ticket prices and discounts	**M**	Allows metro authorities to later prices
11. Recording suspicious transactions	**S**	Helps detect fraud
12. Secure payment encryption	**M**	Ensures security
13. Tracking revenue	**S**	Important for keeping records, however can be done manually as well, but that requires a lot of effort
14. Handling system efficiency	**M**	System should work efficiently in heavy load environment

---

Q-3. Identify three different EPICs (or collection of user stories) where the conflicts between the requirements occur? Do you think that the conflicts can be resolved? How?

EPIC 1 :

User stories in conflict →

User story 2 : travelers want fast and flexible payment options

User story 11: security team wants to detect fraudulent activities

User story 12 : payment providers want payments to be encrypted
Here, encryption and fraud detection activities may delay payments. When passengers want more payment options like cash, there is a higher risk of fraud (like fake currency notes) compared to digital payments.

Digital payments could be encouraged by introducing small discounts for opting for them.

EPIC 2 :
User stories in conflict →
User story 5 : travelers want to issue metro passes
User story 6 : frequent travelers want discount
User story 10 : metro authorities want to adjust prices so that revenue targets are met
Here, the provision of discounts could be exploited by multiple people sharing their metro passes. This could also lead to revenue targets not being met.
By offering discounts at off-peak hours and using a biometric or OTP authentication, such issues can be resolved.

EPIC 3 :
User stories in conflict →
User story 9 : passengers want refund if transaction fails
User story 13 : tracking revenue
Travelers want instant refunds whereas finance officers need time to verify and process refunds. If too many refunds are processed in real time, it may slow down the system.
By limiting daily refunds per user, and instant refunds for failed transactions due to system errors this problem can be solved.