REQUIREMENT ANALYSIS

Date	17th June 2025
Team ID	LTVIP2025TMID28953
Project Name	CRM Application for Public Transport Management System
Maximum Marks	

CUSTOMER JOURNEY MAP

This journey map outlines the end-to-end experience of a public transport passenger — from the moment they inquire about bus services to receiving trip and payment confirmations. It also highlights critical touchpoints with the CRM system, demonstrating how automation, validation, and accurate data capture improve passenger service, staff efficiency, and overall operational effectiveness.

Step	Passenger Action (Detailed Scenario & Intent)	System Interaction (Detailed CRM Role & Data Capture)
1	**Inquires about available bus trips**A passenger approaches a counter, visits a kiosk, or accesses an online portal to ask about available buses, routes, timings, and fares. Their intent is to find suitable transport options.	**Trip records filtered by Route and Bus Availability**The staff uses the CRM interface to search the *Trip__c* object for active records matching the desired *Route Name__c*. Associated *Bus__c* data (e.g., Model, Capacity) and *Ticket Fare__c* (retrieved via Flow based on route and model) are displayed. This enables real-time trip suggestions and fare transparency.
2	**Selects a route and confirms booking**After evaluating options, the passenger selects a bus trip (e.g., Hyderabad to Warangal - Super Deluxe) and requests a reservation.	**Trip record updated with Passenger Count**The staff confirms booking by updating the *Passenger_Count__c* field in the Trip__c record. Validation rules automatically ensure the selected bus model (e.g., Super Deluxe) does not exceed its capacity (Bus__r.Capacity__c). Errors are flagged in real time to prevent overbooking.
3	**Provides contact and personal details**For record-keeping or further communication, the passenger provides basic details (name, phone number). Their intent is to complete the booking formally.	**Customer or booking-related data captured**Though optional in public bus services, if implemented, a *Passenger__c* object or a related Contact record is created/linked. Phone number validation (using REGEX validation rule) ensures correct format (starts with 6/7/8/9 and is 10 digits).
4	**Receives a printed or SMS/email ticket**The passenger expects confirmation of the trip details including date, time, fare, and seat info. Their intent is assurance of a confirmed seat.	**CRM generates ticket with fare & route info**The system, via a Flow or third-party integration, uses the populated *Trip__c*, *Bus__c*, and *Ticket Fare__c* values to create a ticket (custom object or printable format). Confirmation can be sent via SMS/email if integrated with an email Flow using the stored passenger contact.
5	**Makes payment (if prepaid service)**The passenger pays the fare at the time of booking via cash, card, or UPI. Their intent	**Fare confirmation and tracking via CRM**The *Ticket Fare__c* value for the trip is fetched dynamically using a Record-Triggered Flow based on *Route Name__c* and *Bus Model__c*. The system ensures the correct fare is applied, logged against the trip, and

	is to confirm and lock the seat reservation.	optionally tracked in a *Transaction__c* object.
6	**Receives real-time notifications (optional)**Closer to trip time, the passenger may receive an update on departure, delays, or seat changes. Their intent is to stay informed.	**Optional Notification Flow (Advanced Feature)**If implemented, a Scheduled Flow or Apex Job could trigger notifications based on trip timing or status fields. Email or SMS templates can be dynamically populated with *Trip__c* details and sent to the customer (e.g., 1 hour before departure).
7	**Provides feedback or logs complaints (Post-trip)**After the journey, the passenger might want to give feedback or report issues. Their intent is to share experience and improve service.	**Feedback captured via custom object or form**A *Feedback__c* object can be created linked to the *Trip__c* and *Passenger__c*. CRM can generate reports from this data for service quality dashboards, trend analysis, or future planning.

Summary: CRM Benefits in Each Step

- **Efficiency**: Pre-defined Flows and validation rules eliminate manual errors in bookings and fare calculations.
- **Transparency**: Dynamic ticket fare fetch enhances fare accuracy and builds trust.
- **Scalability**: Built-in roles (Driver, Conductor, Staff) with validation rules and Apex Trigger logic (for verifying Employee roles) ensure data integrity.
- **Communication**: Optional Flows enable proactive alerts and smoother passenger experiences.
- **Insights**: Reports and dashboards built from *Trip__c*, *Employee__c*, and *Ticket Fare__c* data allow management to optimize fleet deployment and route performance.

DATA FLOW DIAGRAM (DFD)

Level 0 – Context Diagram Description

The **Level 0 DFD** (Context Diagram) represents a high-level overview of the entire CRM system for Public Transport Management. It shows how **external users** interact with the system and how information flows between them.

Single Process

- Entire system is depicted as one process: **"Public Transport CRM System"**

External Entities

- **Passenger**: Uses bus services and pays fare.
- **Admin / Depot Manager**: Manages employee, fare, and trip records.
- **System (CRM Platform)**: Salesforce-based CRM handling all backend operations.

Major Data Flows

- Passenger provides journey route, fare is generated.
- Admin adds/updates Bus, Trip, Ticket Fare, and Employee info.
- System stores, processes, and validates trip and fare data.
- CRM auto-sends trip/fare confirmation or rejection emails to Admin or Employee if errors occur.

Level 0 DFD Diagram Representation (Conceptual)

[Passenger] ---------->

(CRM for Public Transport System)

[Admin/Depot Manager] ---------->

(System handles Trip, Fare, Bus, Employee, Ticket)

----------> [Email Notification]

Level 1 – Detailed DFD Description

The Level 1 DFD breaks down the single high-level process into **detailed sub-processes**, objects (data stores), and input/output relations between the components.

Sub-Processes:

1. Create Employee (Driver / Conductor)
2. Add Bus Details
3. Create Trip Record
4. Auto-fetch Ticket Fare
5. Validate Employee Role

6. Generate Fare Reports
7. Send Email Confirmation

Level 1 – DFD Table

Step	Process	Input	Output	Data Store (Object)
1	Create Employee	Employee Info (Name, Role, ID, Type)	Employee__c record	Employee__c
2	Add Bus	Bus Details (Bus No, Model, Type)	Bus__c record	Bus__c
3	Create Trip	Route, Date, Bus, Driver, Conductor	Trip__c record	Trip__c
4	Auto-fetch Fare via Flow	Source + Destination + Bus Model	Ticket Fare auto-filled	Ticket_Fare__c
5	Validate Employee Role	Assigned Driver/Conductor	Valid/Invalid Assignment Message	Employee__c, Trip__c
6	Generate Reports	Daily/Monthly Trip + Fare Data	Fare Reports (route-wise, daily)	Trip__c, Ticket_Fare__c
7	Send Email Notification	Fare Generation / Trip Error	Confirmation Email	Notification Log / Sent Log

Additional Notes:

- **Automation via Flow**:
    - A **record-triggered flow** fetches fare automatically from the Ticket_Fare__c object based on Source, Destination, and Bus Model when a Trip__c is created.
- **Apex Trigger (Validation)**:
    - A trigger checks that the driver and conductor assigned to a trip exist in the Employee__c object and have the correct role type.
- **Email Flow**:
    - Confirmation email is sent when a fare is generated or an employee assignment is invalid.

SOLUTION REQUIREMENTS

Functional Requirements

The functional requirements define what the system **must do** to support the management of public transport operations such as buses, trips, employees, and fare automation.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Employee Management	- Create Employee__c records- Classify as Driver or Conductor- Assign roles via picklist- Track active/inactive status
FR-2	Bus Management	- Create Bus__c records- Capture details like Bus Number, Model, Type, Capacity- Link to route if applicable
FR-3	Route and Trip Management	- Create Trip__c records- Assign Bus, Driver, Conductor- Specify Source, Destination, Date, and Timings
FR-4	Ticket Fare Automation	- Create Ticket_Fare__c object with Source, Destination, Bus Type- Fetch fare automatically via Flow when Trip__c is created
FR-5	Validation and Data Integrity	- Apex Trigger to validate that assigned Driver/Conductor exist in Employee__c and roles are correctly mapped
FR-6	Confirmation Communication	- Send email notification after Trip or Fare creation- Flow fetches recipient email from related object (Admin or Employee)
FR-7	Role-Based Access	- Define custom profiles for Admin, Dispatcher, Employee- Apply Permission Sets to restrict access to sensitive or irrelevant data
FR-8	Report and Dashboard Generation	- Generate reports for Trips per day, Fare by Route, Bus usage, Employee allocation- Create dashboards to monitor operations
FR-9	Data Relationship Management	- Use Lookup relationships between Bus__c, Trip__c, Ticket_Fare__c, and Employee__c to ensure accurate cross-object data mapping
FR-10	Error Handling and Monitoring	- Display custom error messages when invalid data is entered (e.g., Driver not found)- Log rejected fare fetch attempts

Non-Functional Requirements

The non-functional requirements define the **quality attributes** that the CRM system should meet to ensure it is secure, efficient, scalable, and reliable.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The UI must be intuitive for transport staff, dispatchers, and administrators with

		minimal training required.
NFR-2	Security	Enforce access control using Salesforce Profiles and Permission Sets to protect employee, trip, and fare data.
NFR-3	Reliability	Ensure consistent operation of flows, triggers, and validations across all standard trip creation and fare fetch processes.
NFR-4	Performance	System should auto-fetch fare, validate employee roles, and generate reports with low latency.
NFR-5	Availability	The system should be operational throughout transport hours (e.g., 5 AM to 11 PM), with minimal downtime.
NFR-6	Scalability	Capable of handling increasing numbers of buses, routes, trips, and employees as the transport network expands.
NFR-7	Maintainability	Automation tools (Flows, Triggers, Validation Rules) should be easily editable and configurable by administrators as operational policies evolve.
NFR-8	Auditability	Field history tracking should be enabled on key fields (e.g., Employee Role, Trip Assignment, Fare Amount) to track changes for accountability.

TECHNOLOGY STACK

The **CRM Application for Public Transport Management System** was developed on the **Salesforce platform**, leveraging its robust low-code and pro-code capabilities to automate public transport operations such as trip scheduling, fare management, bus tracking, and employee role assignment. The system integrates custom data modeling, backend automation, real-time validations, reports, dashboards, and secure access control, ensuring reliability, scalability, and efficiency.

Category	Tools/Technologies Used	Explanation
Platform	Salesforce Lightning Experience	The entire system is developed using Salesforce Lightning UI, which enhances user experience through a fast, responsive, and component-based interface.
Automation	Record-Triggered Flows, Scheduled Flows	- Flows automate critical tasks such as fare fetching based on route and bus type, and email notifications for trip assignments.
Scripting	Apex Triggers and Apex Classes	Apex is used to validate that only eligible employees (Driver/Conductor) are assigned to trips and enforce role-specific logic.
Data Modeling	Custom Objects: Bus__c, Trip__c, Ticket_Fare__c, Employee__c	Custom objects represent operational units such as buses, trips, fares, and staff. Lookup relationships link these entities for accurate data mapping.
Validation & Rules	Validation Rules, Formula Fields	- Validation rules ensure correct assignments (e.g., only drivers can be assigned as drivers).- Formula fields calculate fares and trip info.
Communication	Email Alerts, Record-Triggered Flows, Email Templates	Automated flows send trip assignment confirmations and fare update emails using pre-designed templates, ensuring timely communication.
Reporting & Insights	Custom Reports, Dashboards	Reports analyze trip frequency, bus utilization, and fare data. Dashboards visualize performance metrics such as trips per route and employee allocation.
Access Control	Profiles, Permission Sets	Profiles and Permission Sets define what data and functionality is accessible to roles like Admin, Dispatcher, Driver, and Conductor.

Why This Stack Was Chosen:

- **Salesforce Lightning** provides a fast, modular UI suitable for managing complex transport data.
- **Apex scripting** enables enforcement of specific transport policies (e.g., employee-role mapping).

- **Flows and Email Automation** streamline backend operations such as fare auto-fetching and trip notifications.
- **Reports and Dashboards** allow stakeholders to monitor key KPIs like daily trips, fare collection, and bus usage.
- **Validation Rules and Formula Fields** ensure real-time data integrity and calculated accuracy in fare and trip assignments.

Sample Tools Used in Development:

- **Object Manager** – for defining custom fields and relationships across Bus, Trip, Fare, and Employee objects.
- **Flow Builder** – to create automation for fare calculation and email notifications.
- **Apex Developer Console** – for writing and testing Apex Triggers to enforce employee role logic.
- **Email Template Builder** – for designing professional trip assignment and fare summary messages.
- **Report Builder** – to generate analytical and operational reports (e.g., Trips per Bus, Fare by Route).
- **Setup Menu (Profiles/Permission Sets)** – to restrict access and enable role-based usage of features.