

# A CRM Application For Public Transport Management System



---

# FINAL PROJECT REPORT

**Project Title:** A CRM Application for Public Transport Management System

---

## Team Information

Team ID	LTVIP2025TMID29803	Email ID
Team Size	4	
Team Leader	Ganapathiraju Jayanth Varma	Jayanthalganapathi76@gmail.com
Team Member	Boyina Purneswar	Purneswarboyina@gmail.com
Team Member	Bharath Adithya Nakka	bharathadithya.nakka@gmail.com
Team Member	Ch Leela Kalyani	Leelakalyanich@gmail.com

---

## 1. INTRODUCTION

**Building a Centralized CRM for Efficient Public Transport Management**

### 1.1 Project Overview – Transforming RTC Operations through CRM Technology

In an era of growing urbanization and mobility needs, public transport systems are expected to deliver efficient, timely, and accountable services. However, many Regional Transport Corporations (RTCs) still rely on outdated, manual processes for managing their daily operations. These challenges include inconsistent employee management, error-prone fare calculation, and the lack of centralized reporting systems.

To address these gaps, this project introduces a custom-built CRM application using the Salesforce platform. Designed specifically for RTCs, this solution digitizes critical processes such as employee role assignment, trip creation, ticket fare calculation, and data reporting.

It harnesses Salesforce's automation tools, custom object modeling, and analytics capabilities to create an intelligent, scalable, and maintainable system.

### **Key Features of the Solution:**

- Customized object model supporting buses, bus stations, employees, trips, and fares
- Automated logic for fare calculation and employee role validation using flows and Apex triggers
- Formula-based fields for real-time computation of trip fare, employee age, and more
- Comprehensive dashboards and reports for administrative decision-making
- Centralized control with accurate data relationships and validations

### **1.2 Purpose – Solving Operational Inefficiencies in RTCs**

The purpose of this project is to address the real-world problems faced by RTCs due to fragmented workflows and manual data entry. By implementing a centralized CRM system using Salesforce, the project aims to:

- Streamline operations through automated data flows
- Improve data accuracy using built-in validation rules and relationships
- Enable informed decision-making through dynamic reporting tools
- Create a system that is flexible, user-friendly, and scalable for future integration

## 2. Ideation Phase

---

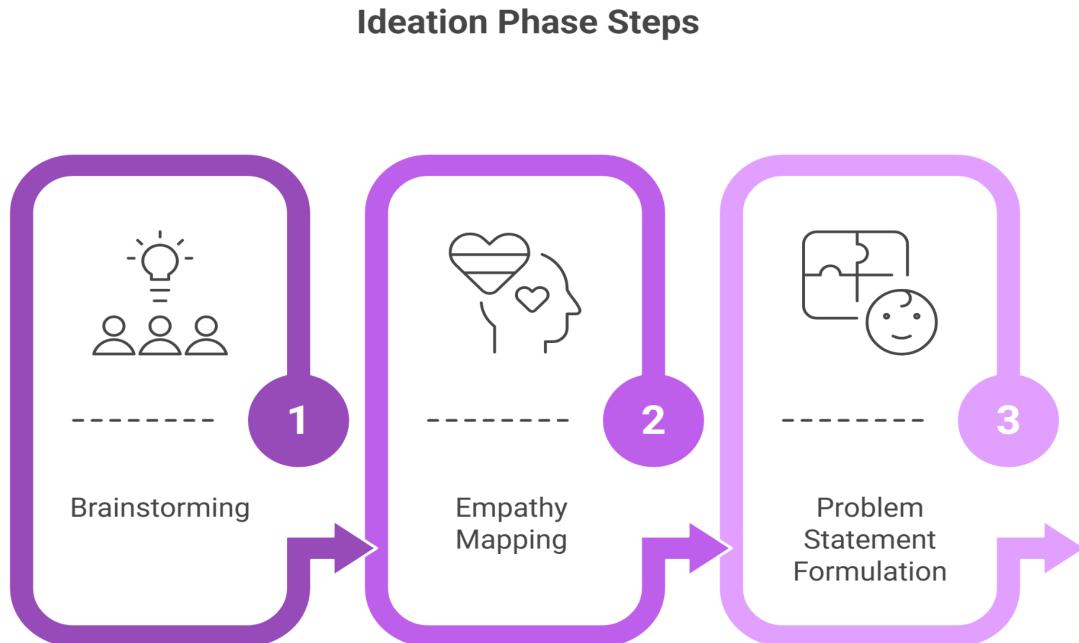
The Ideation Phase serves as the foundation of any successful project. It blends creativity, user empathy, and structured thinking to identify the core problem, generate meaningful ideas, and prioritize solutions that bring value to users. Where creativity and structured thinking combine to find meaningful and impactful documentation.

In our project titled: “A CRM Application for Public Transport Management System”. This phase was critical to ensure we built a system tailored to the actual challenges faced by RTC departments, bus station managers, conductors, and passengers.

---

The ideation phase included three main steps:

1. Brainstorming
2. Empathy Mapping
3. Problem Statement Formulation



---

## **2.1 Brainstorming & Idea Prioritization Template**

---

### **Step 1: Team Gathering, Collaboration, and Selecting the Problem Statement**

Our team convened with the goal of identifying inefficiencies in existing RTC systems and proposing a tech-driven solution using Salesforce. Through collaborative meetings, online whiteboards, and use-case discussions, we collectively explored pain points faced by public transport staff and administrative heads. We reviewed real-world operations and identified that most RTC systems rely heavily on manual workflows for managing:

- Bus trip schedules and fares
- Driver and conductor assignments
- Ticket fare collection
- Monthly performance reporting

After several discussions, we clearly defined the core issue:

#### **Problem Statement:**

"RTC departments lack a unified digital platform for managing buses, employees, ticketing, and operational metrics in real time. Existing manual processes are inefficient, error-prone, and restrict access to performance insights."

This became the backbone of our project scope.

---

## Step 2: Brainstorm, Idea Listing, and Grouping

We performed a team-wide brainstorming session using a digital board where everyone contributed raw ideas. The ideas were categorized into themes:

- **Data Management:** centralized employee, bus, and trip data
- **Automation:** real-time fare calculation, role-based triggers
- **Reporting:** monthly dashboards for trip count, passenger data, and revenue
- **Validation & Access Control:** rule-based data integrity and secure access

From around 25–30 ideas, we grouped and shortlisted the ones that aligned directly with operational efficiency.

---

## Step 3: Idea Prioritization

Each grouped idea was evaluated on:

- **Feasibility:** How easily it could be implemented on Salesforce
- **Impact:** The significance of the feature on transport operations
- **Urgency:** Whether it solved a current, pressing problem

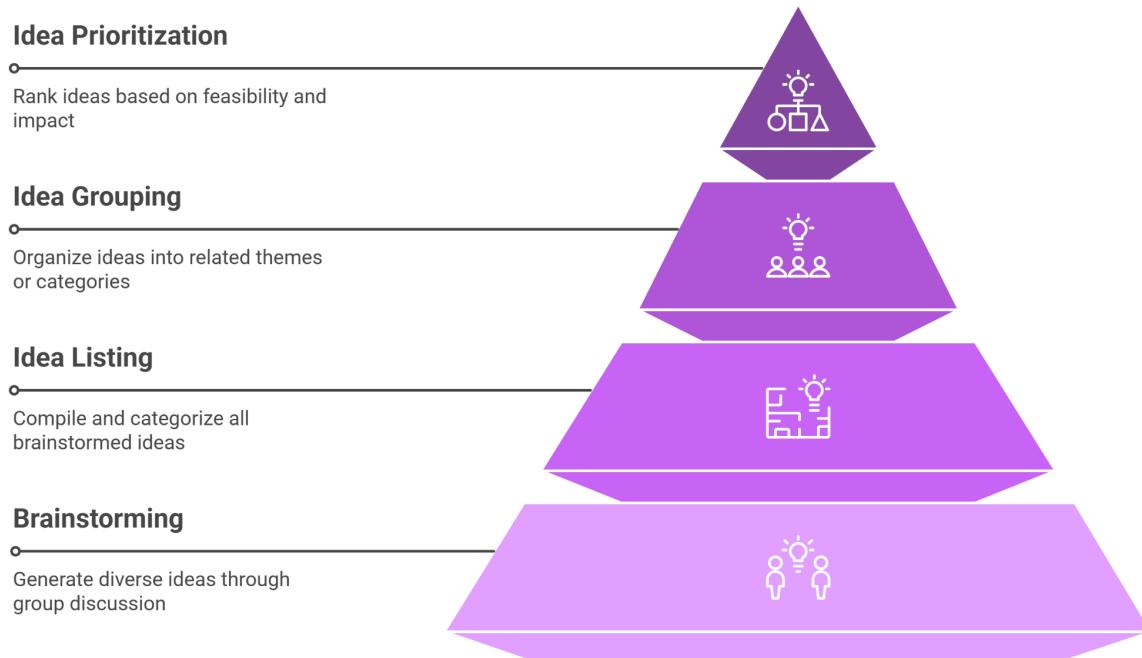
We created a decision matrix that helped us arrive at an **MVP** (Minimum Viable Product) plan:

- **Top Priority Features:**

- Automated fare and passenger data updates
- Role-based employee assignment (Driver/Conductor)
- Trigger-based alerts for invalid assignments
- Real-time summary dashboards (e.g., trips, passengers, revenue)
- Controlled and dependent picklists for bus and route management

These features formed the scope of our system design in the later phases.

**Idea Prioritization Pyramid**



---

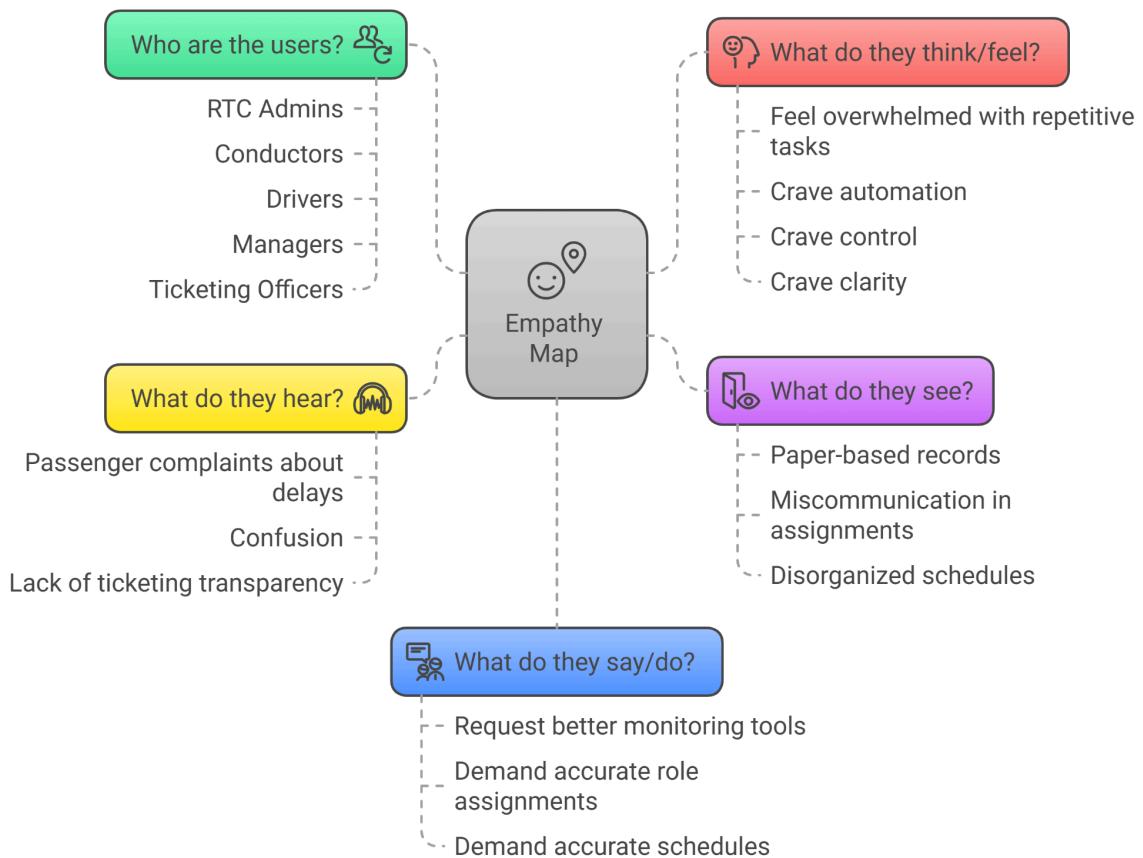
## 2.2 Empathy Mapping- Empathize & Discover

---

### Empathy Map Canvas

An empathy map is a visual tool that helps teams deeply understand their users' experiences, pains, and expectations. We used it to map the daily journey of RTC staff, including administrators, bus station managers, and field employees.

Empathy Map for RTC Staff



By stepping into the user's shoes, we ensured that our Salesforce CRM features (formulas, flows, triggers, dashboards) directly addressed their key frustrations.

---

## 2.3 Define the Problem Statements

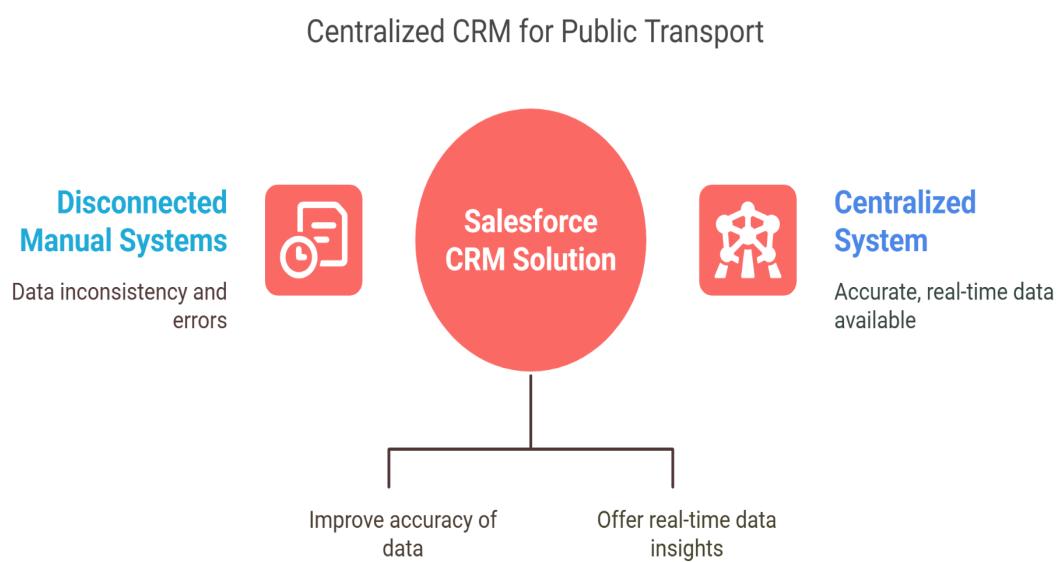
---

### Customer Problem Statement Template

To build a successful solution, it's essential to clearly define what the customer/user is struggling with. This helped us stay focused on delivering real-world impact instead of just implementing technical features.

#### Final Customer Problem Statement:

RTC departments manage public transportation manually using disconnected systems. This leads to data inconsistency, assignment errors, revenue loss, and lack of real-time performance visibility. A centralized Salesforce CRM system can digitize workflows, ensure accuracy, and provide actionable insights through dashboards, flows, and automation. This statement aligns with the end user's expectations and guided our object design, validations, formulas, triggers, and reports.



# 3. Requirement Analysis

---

The Requirement Analysis Phase focuses on collecting, organizing, and validating everything your system needs to accomplish. It ensures that the solution you're building is not only technically sound but also directly aligned with what your stakeholders (RTC departments, drivers, managers) truly need. In our project, "A CRM Application for Public Transport Management System," this phase helped bridge the gap between problem understanding and system design, using real user journeys, data flows, and solution requirements. Where user-centric planning meets technical clarity to transform a problem into a buildable solution.

---

## 3.1 Customer Journey Map-Understanding User Experience Flow

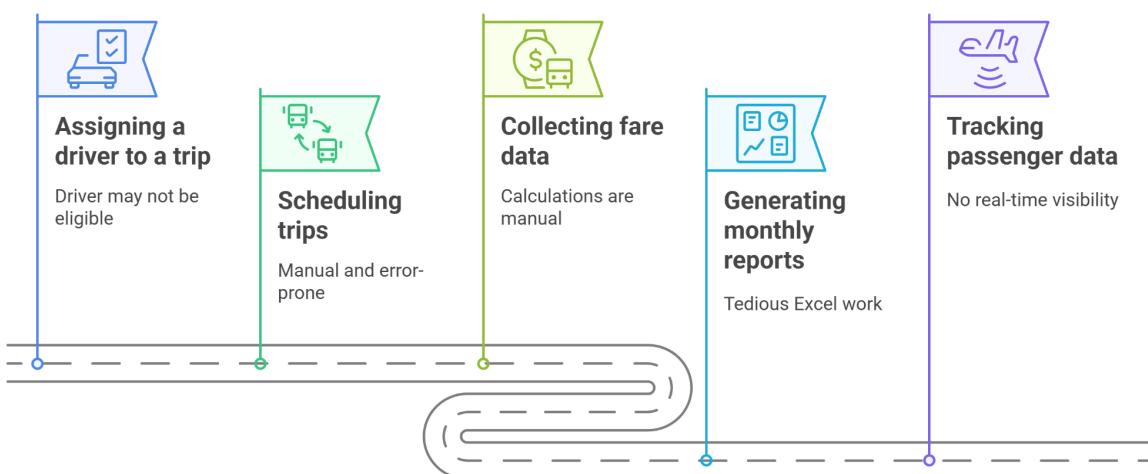
---

**Purpose:** The **Customer Journey Map** visualises how RTC employees (e.g., Bus Station Managers, Admins, Drivers, Conductors) interact with the transport system daily. It highlights key actions, pain points, and opportunities for improvement, guiding CRM design from a real-world usage perspective.

---

### Journey Steps (For an RTC Admin):

RTC Employee Interaction with Transport System



This journey helped prioritize features like formula fields, validation rules, and real-time dashboards in our object and flow design.

---

## 3.2 Data Flow Diagram

---

**Purpose:** Mapping Information Flow Between Objects

The Data Flow Diagram (DFD) models how information moves between Salesforce objects and components in the CRM. It helped us structure relationships between:

- Bus Station, Bus, Trip, Employee, and Ticket Fare

### Level 1 DFD Overview:

#### 1. Admin Inputs:

- Adds bus, station, trip details
- Assigns drivers and conductors (from Employee object)

#### 2. System Logic:

- Validates employee role (Driver or Conductor)
- Fetches fare via flow based on route + model
- Calculates total fare using formula (Passenger Count × Fare)

#### 3. Outputs:

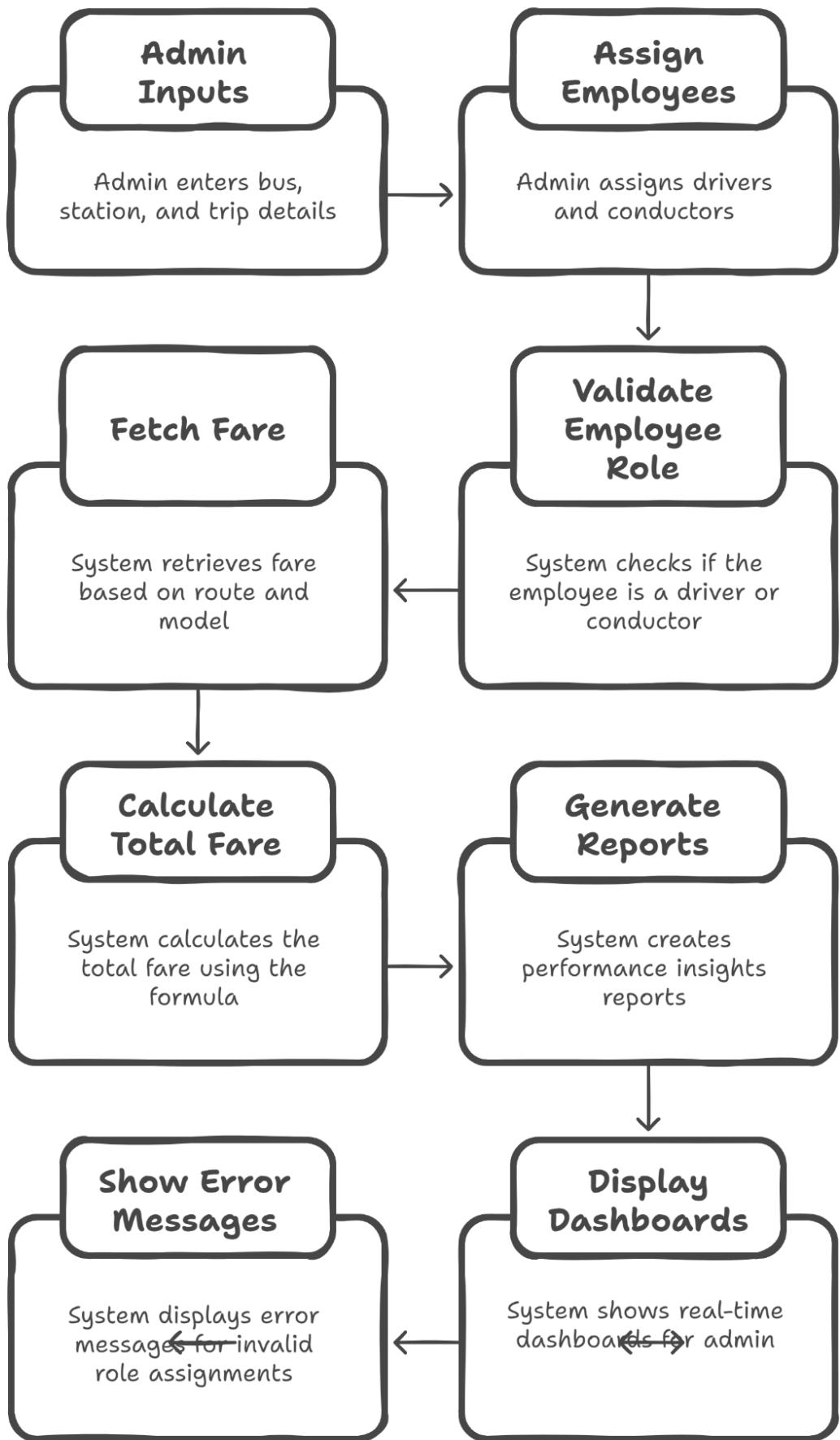
- Performance insights via reports and Real-time dashboards for admin
- Error messages for invalid role assignments

---

## Objects Involved in Bus Transportation Process

Bus Station → Bus → Employee → Trip → Ticket Fare

## Level 1 DFD Overview



---

## **3.3 Solution Requirements**

---

### **What the CRM System Must Do**

#### **Functional Requirements:**

- Create custom objects for: **Bus Station, Bus, Trip, Employee, Ticket Fare**
- Establish relationships using lookup and formula fields
- Validate employee roles using Apex Triggers
- Automate fare fetching via Flows
- Support performance dashboards & summary reports

#### **Non-Functional Requirements:**

- User-friendly Lightning App Interface
- Real-time field-level validation
- Centralized database with accurate relationships

---

## 3.4 Technology Stack

---

### Tools & Platforms Used for CRM Implementation

Category	Technology Used	Description
Platform	Salesforce Lightning	Used to build custom CRM using standard & custom objects
Automation	Flows & Validation Rules	Automate fare fetching and restrict invalid data
Custom Logic	Apex Triggers & Classes	Used to validate Driver/Conductor ID
Reports & Dashboards	Salesforce Reports	To analyze employee data, trips, revenue, etc.
UI/UX	Lightning App Builder	For creating a unified app view for Public Transport CRM

---

## Summary

This Requirement Analysis Phase helped ensure that all CRM features were grounded in user pain points, supported by data structures, and enabled by the right technologies. It directly shaped how we approached object modeling, data automation, UI design, and performance reporting in Salesforce.

---

# 4. Project Design Phase

---

The Project Design Phase defines the logical, technical, and functional foundation of the solution. It ensures that your proposed Salesforce CRM not only solves the right problems but is also scalable, maintainable, and aligned with Salesforce architecture principles. Where validated problems transform into structured, scalable, and implementable solutions.

In our project, “A CRM Application for Public Transport Management System,” this phase bridges the gap between ideation and execution by converting insights from the previous requirement analysis into a well-structured CRM solution.

---

## 4.1 Problem–Solution Fit

---

### Problem Recap:

RTC (Regional Transport Corporations) operate in a domain that requires real-time tracking of buses, trips, employees, and fare collection. However, operations are largely manual or siloed in spreadsheets, causing:

- Data entry errors and redundancy
  - No real-time visibility for decision-makers
  - Poor assignment of drivers/conductors
  - Difficulty in fare management and trip reporting
-

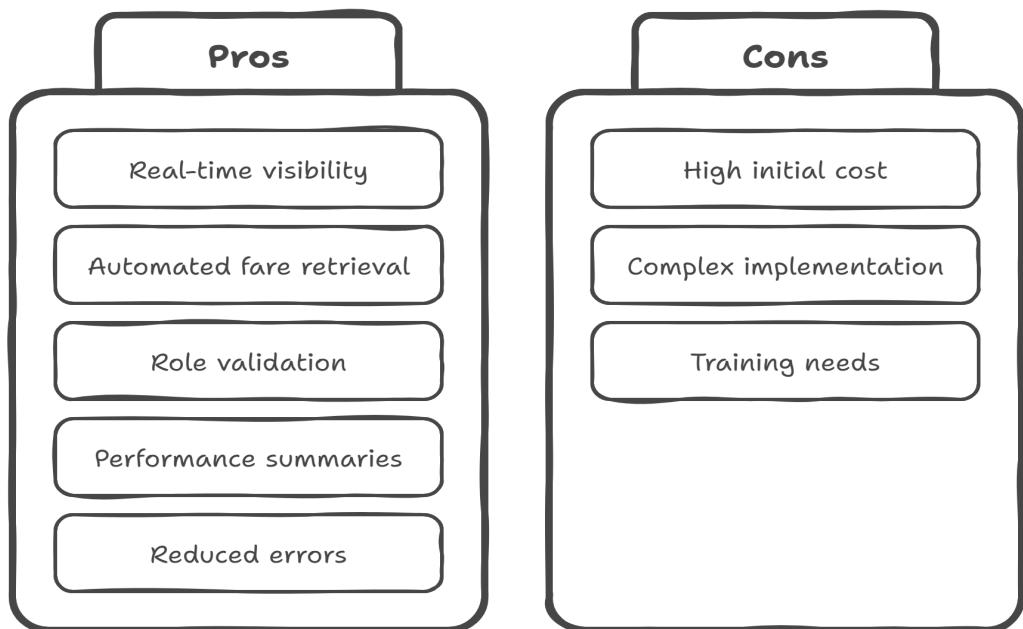
## Does the Proposed Solution Fit?

Yes. The Salesforce CRM solution:

- Introduces **object-level control** for Buses, Stations, Employees, Trips, and Ticket Fares
- Enables **automated fare retrieval** through Flows
- Validates driver and conductor roles using **Apex Triggers**
- Summarizes performance using **Reports and Dashboards**
- Uses formula fields to reduce calculation errors
- Creates centralized views using **Lightning App Builder**

Thus, it directly fits the core operational pain points of RTC workflows.

## Salesforce CRM for RTC



---

## 4.2 Proposed Solution

### How Our CRM Will Solve the Identified Problems

---

Our proposed CRM application is designed to digitize and streamline RTC operations using Salesforce's declarative and programmatic capabilities.

#### Key Functional Features:

- **Custom Objects:**

- Bus Station, Bus, Trip, Ticket Fare, Employee

- **Automation & Validation:**

- Role verification (Driver/Conductor) via **Apex Triggers**
  - Fare calculation automation via **Flows**
  - Input control via **Validation Rules**

- **Formula Fields for Efficiency:**

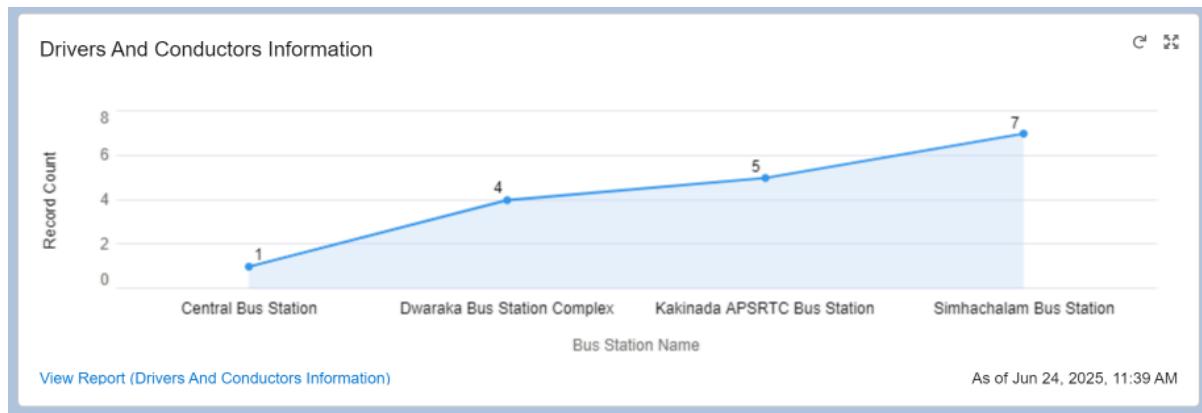
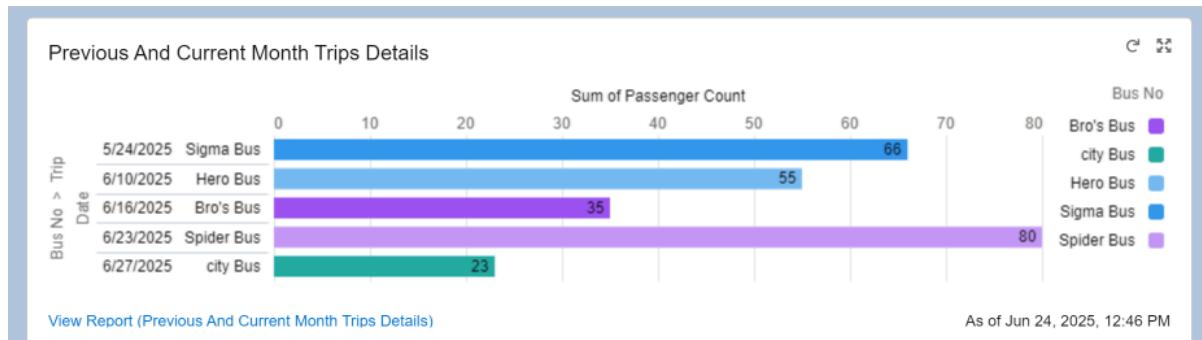
- Age, Experience, Date of Retirement for employees
  - Total Fare = Passenger Count × Ticket Fare
  - Driver & Conductor Names (auto-derived via lookups)

- **UI & Navigation:**

- Public Transport App using **Lightning App Builder**
  - Tabs for all custom objects

- Page layouts designed by object and role
- **Reports and Dashboards:**
  - Trip Summary Reports
  - Driver/Conductor Assignment Reports
  - Monthly Revenue and Passenger Count Dashboards

Together, these components will offer a centralized, role-driven, and insight-ready system.



---

## 4.3 Solution Architecture

Visualizing the Technical Structure and Object Relationships

---

### Object Relationship Overview:

Object	Key Fields / Features
<b>Bus Station</b>	Name, Category, Amenities, Address
<b>Bus</b>	Linked to Bus Station, Category, Model, Capacity
<b>Employee</b>	Name, Role (Picklist), DOB, Experience, Lookup to Station
<b>Trip</b>	Linked to Bus, Driver, Conductor, Ticket Fare, Date, Passenger Count, Total Fare (Fx)
<b>Ticket Fare</b>	Route, Bus Model, Fare

### Lookups:

- Employee → Bus Station
- Trip → Bus, Employee (Driver), Employee (Conductor), Ticket Fare
- Bus → Bus Station

---

## **Formulas:**

- Total\_Amount\_\_c = Passenger\_Count\_\_c \* Ticket\_Fare\_\_c
  - Driver\_Name\_\_c = Driver\_Id\_\_r.Employee\_Name\_\_c
  - Conductor\_Name\_\_c = Conductor\_Id\_\_r.Employee\_Name\_\_c
- 

## **Automation:**

- Flows for fare fetch logic
  - Triggers for role validation
  - Reports & dashboards for output
- 

## **Summary**

The Project Design Phase ensured that our CRM not only met the users' needs but also followed Salesforce best practices in object modeling, validation, automation, and user experience. This clear blueprint guided our execution in upcoming development and configuration phases.

---

# 5. Project Planning Phase

---

The Project Planning Phase converts high-level milestones into actionable sprints aligned with the internship timeline. This helps streamline delivery and keeps all team members aligned with progress, ownership, and deadlines. Structured task breakdown and time-bound execution planning ensures delivery efficiency.

---

## 5.1 Project Planning Template

### Sprint Schedule – Based on Project Milestones

Sprint	Functional Requirement (Epic)	Task (Mapped from Milestone)	Priority	Team Members
Sprint-1	Developer Setup & Basic Objects	Creating Developer Account & Activating Org	High	Member 1
Sprint-1	Custom Object Creation	Creating custom objects – Bus, Bus Station, Employee, Trip, Ticket Fare	High	Member 1, 2
Sprint-2	UI Tabs & App Creation	Creating Tabs & Lightning App	High	Member 3
Sprint-2	Field Configuration	Creating fields, formula fields, picklists, relationships	Medium	Member 1, 3
Sprint-3	Layouts & Validations	Page Layouts + Validation Rules	High	Member 2, 4
Sprint-3	Flows & Triggers	Automations using Flows and Apex Triggers	High	Member 2, 3

Sprint-4	Reports & Dashboards	Generate Reports and create Dashboards	High	Member 4
Sprint-4	Final Integration & Conclusion	Final Review, Testing, and Functional Summary	Medium	All Members

---

## Project Tracker & Sprint Timeline

Duration: Each sprint is 6 days, aligned with your **June 2025 internship schedule**

Sprint	Duration	Sprint Start Date	Sprint End Date	Sprint Release Date
Sprint-1	6 Days	03 Jun 2025	08 Jun 2025	08 Jun 2025
Sprint-2	6 Days	09 Jun 2025	14 Jun 2025	14 Jun 2025
Sprint-3	6 Days	15 Jun 2025	20 Jun 2025	20 Jun 2025
Sprint-4	6 Days	21 Jun 2025	26 Jun 2025	26 Jun 2025

---

## Summary

The **Project Planning Phase** allowed our team to convert 12 major milestones into 4 streamlined sprints with assigned priorities and contributors. By aligning sprints with real internship dates and breaking tasks down into functional chunks, we ensured steady progress and simplified execution.

---

## **6. Project Executable Files**

---

This phase outlines the actual Salesforce configurations, data, and outcomes used and generated during the execution of your project: "A CRM Application for Public Transport Management System." It ensures that all key project elements—objects, data, and output—are traceable and reusable for future reference or assessment. Where practical configurations and working modules of the project are documented for clarity, replication, and validation.

---

### **6.1 Project Files**

#### **Project Executable Files**

The following project files were executed in the Salesforce Developer Org:

---

-  **Milestone 1: Developer Account Setup**
-  **Milestone 2: Object Creation**
-  **Milestone 3: Tab Creation**
-  **Milestone 4: Lightning App Setup**
-  **Milestone 5: Field Creation**
-  **Milestone 6: Page Layouts**
-  **Milestone 7: Validation Rules**
-  **Milestone 8: Flow Setup**
-  **Milestone 9: Apex Trigger**
-  **Milestone 10: Reports**
-  **Milestone 11: Dashboards**
-  **Milestone 12: Final Review**

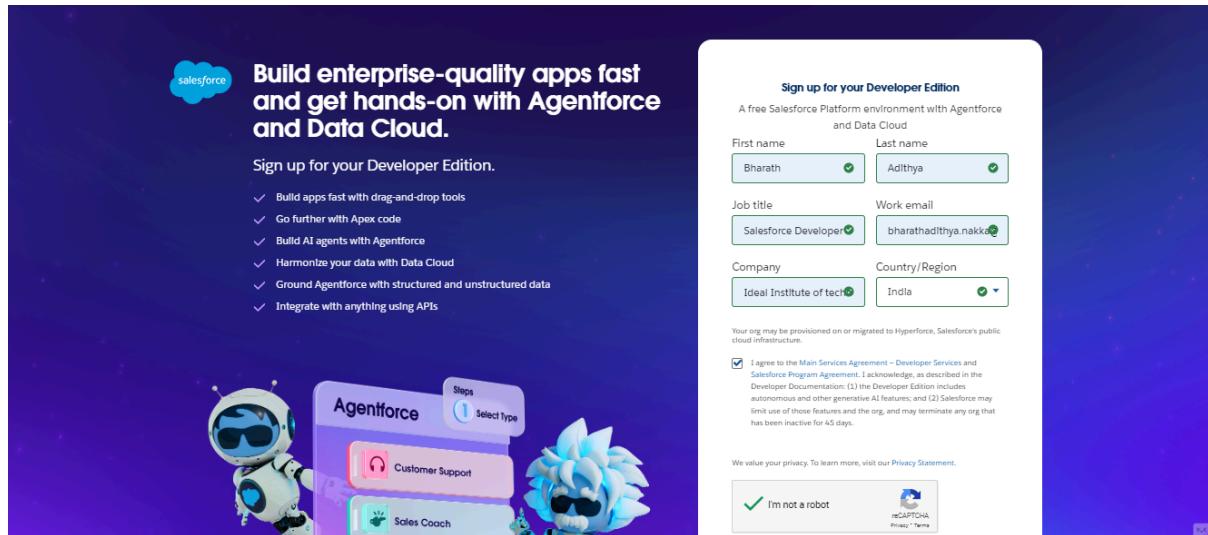
## List of Milestone Tasks with Supporting Screenshots and Descriptions

---

### 📁 Milestone 1: Developer Account Setup

- Created and activated a Salesforce Developer Org.
  - Link: <https://developer.salesforce.com/signup>
  - Setup the base environment for CRM development.
  - Verified access to Object Manager, Flow Builder, and App Builder.
- 

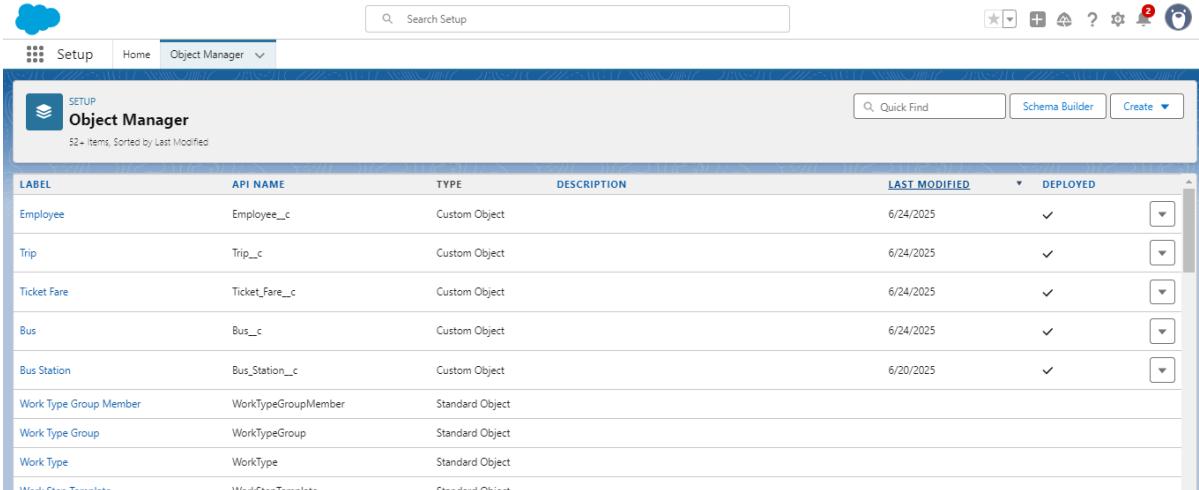
## OUTPUT SCREENSHOT



## Milestone 2: Object Creation

- 
- Created 5 custom objects:
    - Bus Station, Bus, Employee, Trip, and Ticket Fare.
  - Established foundational schema for RTC data tracking.
  - Configured relationships using lookup fields.
- 

### OUTPUT SCREENSHOT



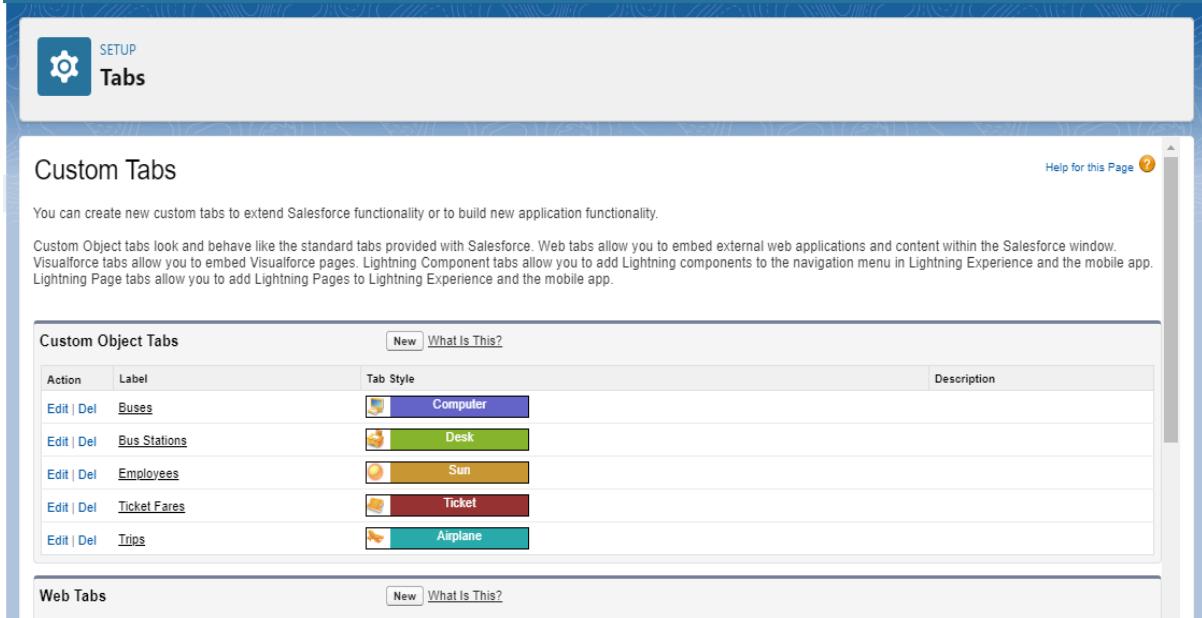
The screenshot shows the Salesforce Object Manager page. At the top, there are tabs for Setup, Home, and Object Manager. A search bar labeled "Search Setup" is followed by various navigation icons. The main area is titled "Object Manager" and displays a table of objects. The table has columns for Label, API Name, Type, Description, Last Modified, and Deployed. The objects listed are:

Label	API Name	Type	Description	Last Modified	Deployed
Employee	Employee_c	Custom Object		6/24/2025	✓
Trip	Trip_c	Custom Object		6/24/2025	✓
Ticket Fare	Ticket_Fare_c	Custom Object		6/24/2025	✓
Bus	Bus_c	Custom Object		6/24/2025	✓
Bus Station	Bus_Station_c	Custom Object		6/20/2025	✓
Work Type Group Member	WorkTypeGroupMember	Standard Object			
Work Type Group	WorkTypeGroup	Standard Object			
Work Type	WorkType	Standard Object			
Work Step Template	WorkStepTemplate	Standard Object			

## Milestone 3: Tab Creation

- 
- Created tabs for each custom object.
  - Enabled easy navigation and object access in the app.
  - Ensured users can create/view records from the UI.
- 

### OUTPUT SCREENSHOT



The screenshot shows the Salesforce Setup interface under the 'Tabs' section. The top navigation bar includes 'SETUP' and 'Tabs'. Below the header, a section titled 'Custom Tabs' provides information about creating new custom tabs. A table lists five custom object tabs:

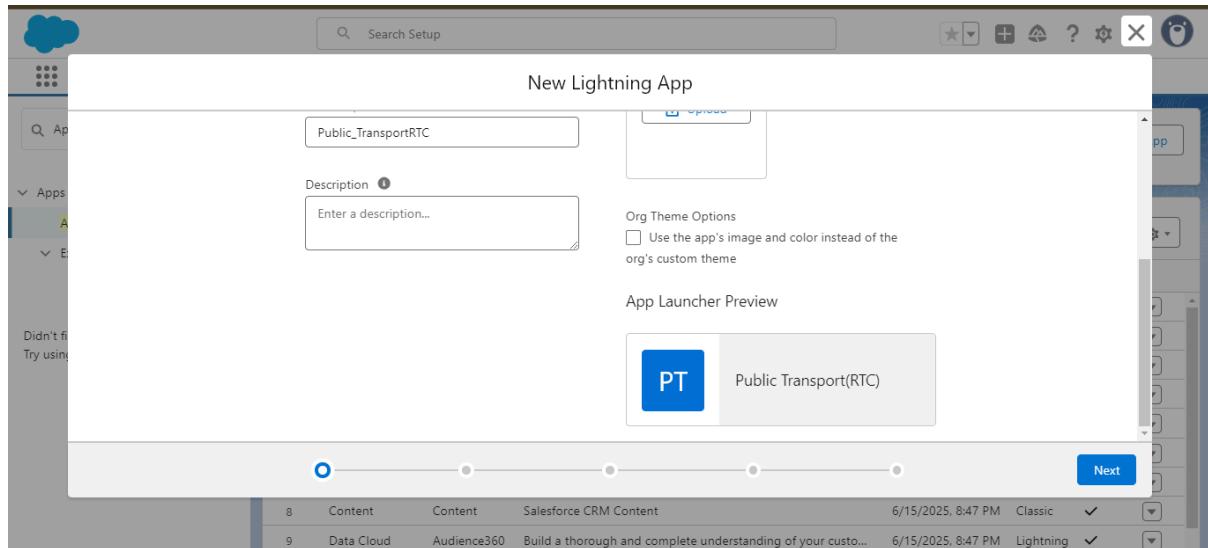
Action	Label	Tab Style	Description
Edit   Del	Buses	Computer	
Edit   Del	Bus Stations	Desk	
Edit   Del	Employees	Sun	
Edit   Del	Ticket Fares	Ticket	
Edit   Del	Trips	Airplane	

Below the table, another section titled 'Web Tabs' is visible.

## Milestone 4: Lightning App Setup

- 
- Built a custom Lightning App named "**Public Transport RTC**".
  - Added relevant tabs to centralize operations.
  - Simplified user workflow by grouping features.
- 

## **OUTPUT SCREENSHOT**

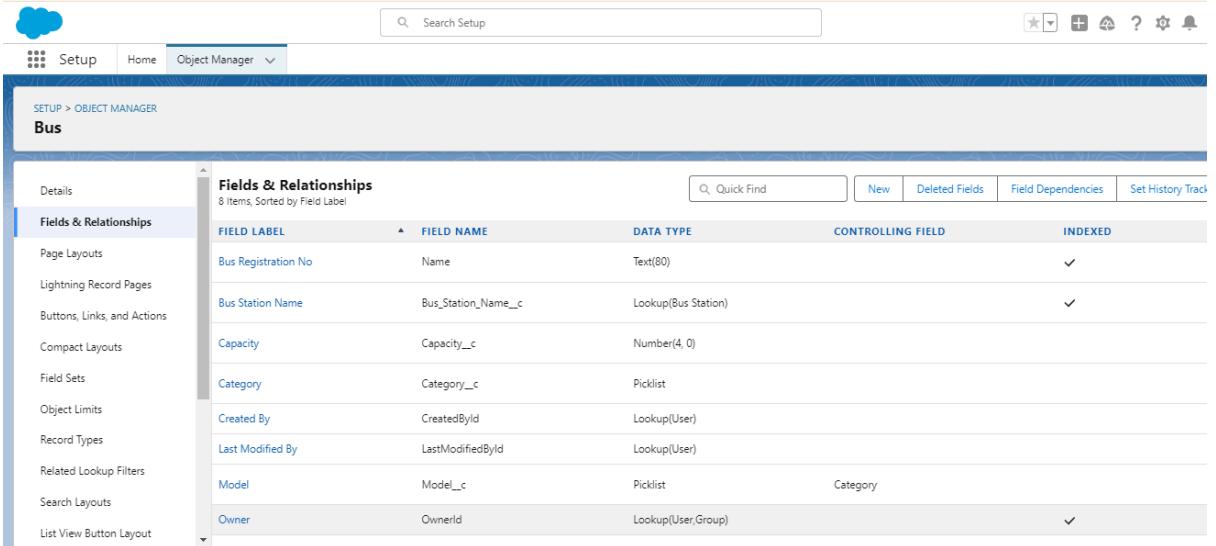


## Milestone 5: Field Creation

- Added custom and formula fields like:
  - Role (Picklist), Age, Experience, Fare, Passenger Count.
  - Implemented: **Total Fare = Fare × Passenger Count** formula.
- Linked Driver and Conductor to Trip via lookup fields.

### OUTPUT SCREENSHOT

#### Bus object Fields



The screenshot shows the Salesforce Object Manager interface for the 'Bus' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The left sidebar lists various setup categories under 'Fields & Relationships'. The main content area displays a table titled 'Fields & Relationships' with 8 items. The table columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Bus Registration No	Name	Text(80)		✓
Bus Station Name	Bus_Station_Name__c	Lookup(Bus Station)		✓
Capacity	Capacity__c	Number(4, 0)		
Category	Category__c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Model	Model__c	Picklist	Category	
Owner	OwnerId	Lookup(User,Group)		✓

## Bus Station object Fields

The screenshot shows the Salesforce Object Manager for the 'Bus Station' object. The left sidebar contains navigation links for Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main area is titled 'Fields & Relationships' and lists 14 items, sorted by Field Label. The fields are:

Field Name	Type	Description
Bus Station Name	Name	Text(80)
Bus Stop Category	Picklist	
City	Text(40)	
Created By	Lookup(User)	
Last Modified By	Lookup(User)	
Last Updated	Formula (Date)	
Location	Text(40)	
Owner	Lookup(User/Group)	
Shelter available	Checkbox	
State/Province	Text(25)	

## Employee object Fields

The screenshot shows the Salesforce Object Manager for the 'Employee' object. The left sidebar contains navigation links for Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main area is titled 'Fields & Relationships' and lists 20 items, sorted by Field Label. The fields are:

Field Name	Type	Description
Date of Retirement	Date	Formula (Date)
Employee Id	Name	Text(80)
Employee Name	Text	Text(80)
Experience	Number	Formula (Number)
Last Modified By	User	Lookup(User)
Owner	User/Group	Lookup(User/Group)
Phone	Phone	Phone
Role	Picklist	
Salary	Currency	Currency(18, 0)
State/Province	Text	Text(18)

## Trip object Fields

The screenshot shows the Salesforce Setup interface under the Object Manager for the 'Trip' object. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area displays the 'Fields & Relationships' section with 21 items. A table lists fields such as Arrival Time, Bus No, Bus Starting Terminal, Conductor Id, Conductor Name, Created By, Departure Time, Destination Terminal, and Driver Id, along with their field labels, data types (e.g., Picklist, Lookup), controlling fields, and indexing status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Arrival Time	Arrival_Time__c	Picklist		
Bus No	Bus_No__c	Lookup(Bus)		✓
Bus Starting Terminal	Bus_Startng_Terminal__c	Text(18)		
Conductor Id	Conductor_Id__c	Lookup(Employee)		✓
Conductor Name	Conductor_Name__c	Formula (Text)		
Created By	CreatedBy	Lookup(User)		
Departure Time	Departure_Time__c	Picklist		
Destination Terminal	Destination_Terminal__c	Text(18)		
Driver Id	Driver_Id__c	Lookup(Employee)		✓

## Ticket fare object Fields

The screenshot shows the Salesforce Setup interface under the Object Manager for the 'Ticket Fare' object. The left sidebar lists various setup categories. The main content area displays the 'Fields & Relationships' section with 6 items. A table lists fields such as Bus Model, Created By, Last Modified By, Owner, Route Name, and Ticket Fare, along with their field labels, data types (e.g., Picklist, Lookup), controlling fields, and indexing status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Bus Model	Bus_Model__c	Picklist		
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Route Name	Name	Text(80)		✓
Ticket Fare	Ticket_Fare__c	Currency(10, 2)		



## Milestone 6: Page Layouts

---

- Customized page layouts per object.
  - Grouped fields logically for usability.
  - Enhanced record readability and data entry experience.
- 

### OUTPUT SCREENSHOT

- Bus object Page Layouts

The screenshot shows the Salesforce Object Manager interface for the 'Bus' object. The left sidebar lists various tabs: Details, Fields & Relationships, Page Layouts (which is selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main area displays the 'Page Layouts' configuration screen. At the top, there are buttons for Save, Quick Save, Preview As..., Cancel, Undo, Redo, and Layout Properties. A 'Quick Find' field is present. Below these are sections for 'Fields' and 'Sections'. The 'Fields' section contains a table with columns for 'Section' (Blank Space), 'Capacity', 'Model', 'Category', 'Created By', and 'Last Modified By'. The 'Sections' section has a single row labeled 'Blank Space'. At the bottom, there are sections for 'Information', 'System Information', and 'Custom Links', each containing various fields like Bus Registration No, Bus Station Name, Category, Model, Capacity, Created By, Last Modified By, and Owner.

- Bus station object Page Layouts

SETUP > OBJECT MANAGER  
Bus Station

**Page Layouts**

Bus Station Detail

Field	Type	Value
Bus Station Name	Text	Sample Text
Amenities	Text	Sample Text
Bus Stop Category	Text	Sample Text
Last Updated	Date	8/24/2025
Street	Text	Sample Text
City	Text	Sample Text
Location	Text	Sample Text
Shelter available	Check	✓
Bench	Check	✓
Owner	Text	Sample Text
StateProvince	Text	Sample Text
ZipPostalCode	Text	Sample Text

- Employee object Page Layouts

SETUP > OBJECT MANAGER  
Employee

**Page Layouts**

Employee Detail

Field	Type	Value
Employee Id	Text	Sample Text
Employee Name	Text	Sample Text
Bus Station Name	Text	Sample Text
Salary	Text	\$123.45
Experience	Text	347.82
City	Text	Sample Text
Date of Birth	Date	8/24/2025
Phone	Text	1415-555-1212
Role	Text	Sample Text
Date of Joining	Date	8/24/2025
Date of Retirement	Date	8/24/2026
Employee Name	Text	Sample Text
Phone	Text	Sample Text
Street	Text	Sample Text
City	Text	Sample Text
StateProvince	Text	Sample Text
Country	Text	Sample Text
ZipPostalCode	Text	Sample Text
Age	Text	639.24
Work Place	Text	Sample Text
Experience	Text	Sample Text
Salary	Text	Sample Text
Created By	Text	Sample Text
Last Modified By	Text	Sample Text

English (United States)  
English (India) keyboard  
To switch input methods, press Windows key+Space.

- Bus object Page Layouts

The screenshot shows the Salesforce Object Manager interface for the 'Trip' object. The left sidebar contains navigation links for Details, Fields & Relationships, Page Layouts (which is selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules, and Object Access.

The main area displays the 'Trip Detail' page layout. It includes sections for Information, Bus Schedule, Passenger Information, and System Information. The 'Information' section contains fields for Trip No, Trip Date, Bus No, Driver Id, and Driver Name. The 'Bus Schedule' section contains fields for Route Name, Bus Starting Terminal, Departure Time, and No. of Stops. The 'Passenger Information' section contains a field for Passenger Count. The 'System Information' section contains fields for Last Modified By, Owner, and Total Amount.

- Bus object Page Layouts

The screenshot shows the Salesforce Object Manager interface for the 'Ticket Fare' object. The left sidebar contains navigation links for Details, Fields & Relationships, Page Layouts (selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Partition Rules.

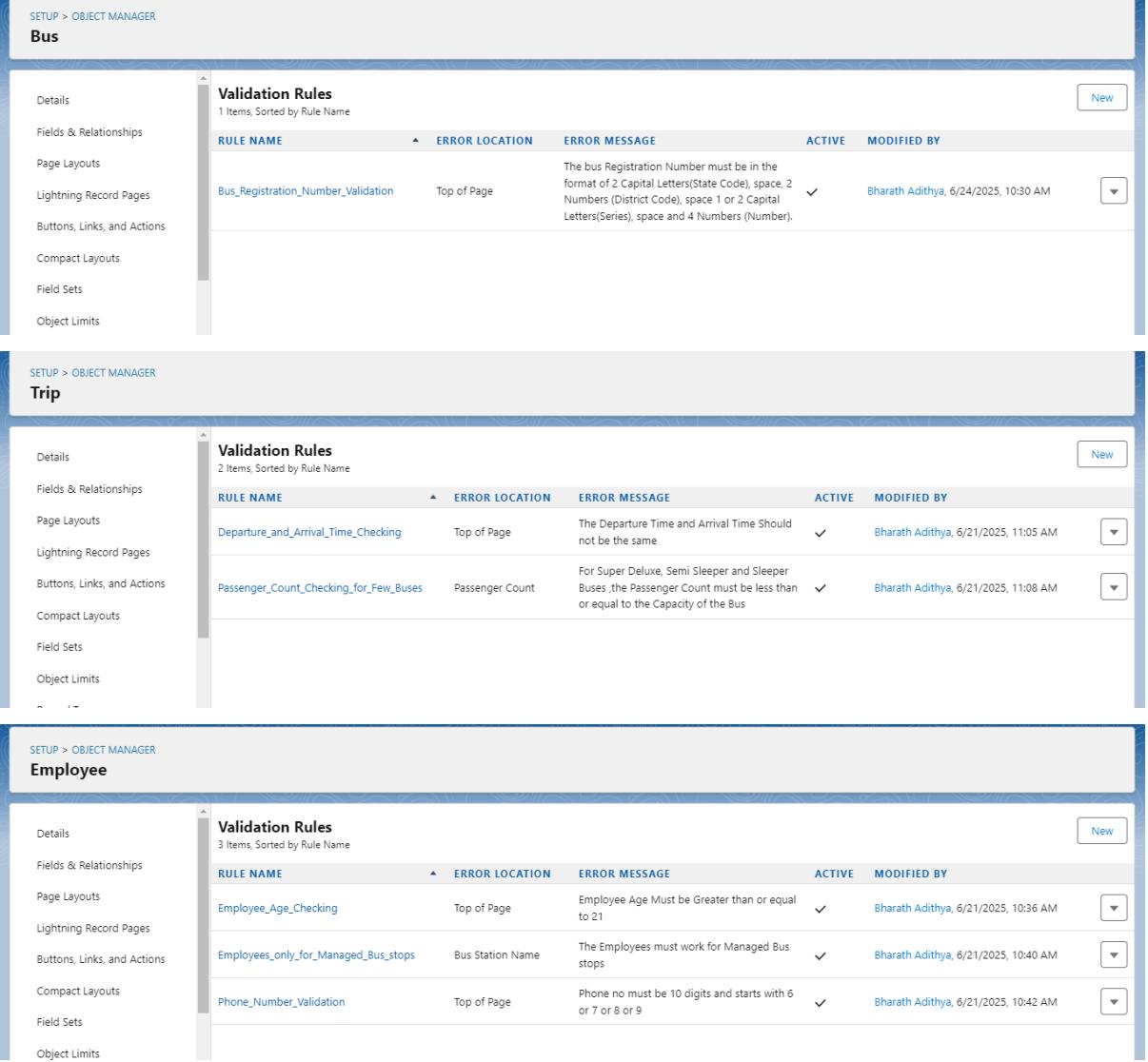
The main area displays the 'Ticket Fare Detail' page layout. It includes sections for Information, System Information, and Custom Links. The 'Information' section contains fields for Route Name, Ticket Fare, and Bus Model. The 'System Information' section contains fields for Created By, Last Modified By, and Owner. The 'Custom Links' section is currently empty.

## Milestone 7: Validation Rules

Enforced data quality with validations like:

- Age  $\geq 21$ , correct phone number format, required fields.
- Prevented invalid inputs and ensured system reliability.

### OUTPUT SCREENSHOT



The image displays three separate screenshots of the Salesforce Object Manager interface, each showing the 'Validation Rules' section for a specific object: Bus, Trip, and Employee.

**Bus Validation Rules:**

Rule Name	Error Location	Error Message	Active	Modified By
Bus_Registration_Number_Validation	Top of Page	The bus Registration Number must be in the format of 2 Capital Letters(State Code), space, 2 Numbers (District Code), space 1 or 2 Capital Letters(Series), space and 4 Numbers (Number).	✓	Bharath Adithya, 6/24/2025, 10:30 AM

**Trip Validation Rules:**

Rule Name	Error Location	Error Message	Active	Modified By
Departure_and_Arrival_Time_Checking	Top of Page	The Departure Time and Arrival Time Should not be the same	✓	Bharath Adithya, 6/21/2025, 11:05 AM
Passenger_Count_Checking_for_Few_Buses	Passenger Count	For Super Deluxe, Semi Sleeper and Sleeper Buses ,the Passenger Count must be less than or equal to the Capacity of the Bus	✓	Bharath Adithya, 6/21/2025, 11:08 AM

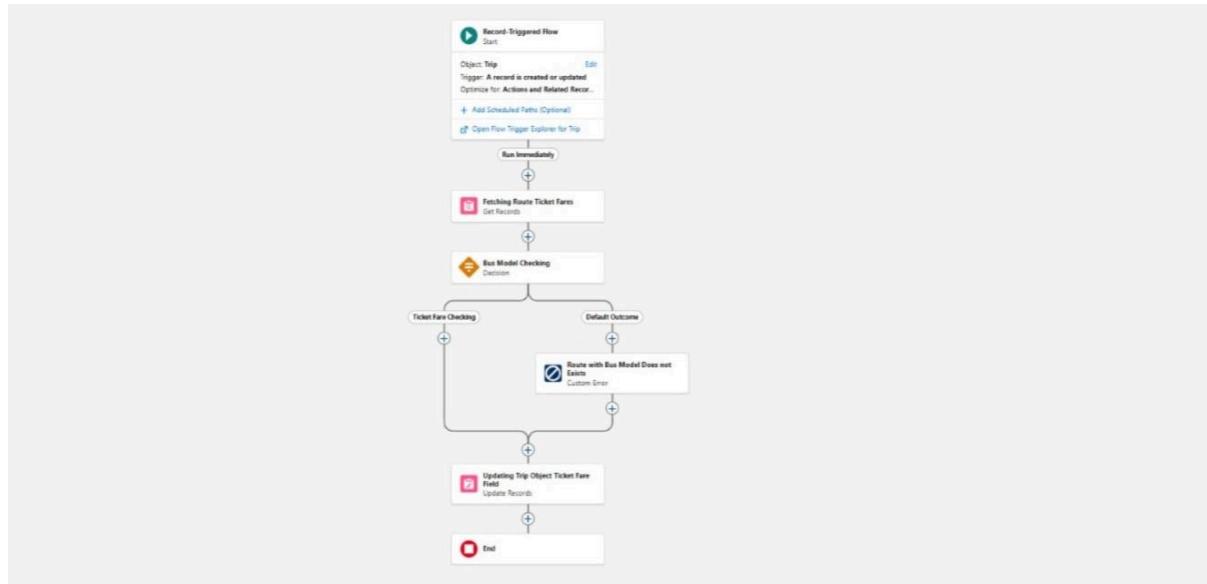
**Employee Validation Rules:**

Rule Name	Error Location	Error Message	Active	Modified By
Employee_Age_Checking	Top of Page	Employee Age Must be Greater than or equal to 21	✓	Bharath Adithya, 6/21/2025, 10:36 AM
Employees_only_for_Managed_Bus_stops	Bus Station Name	The Employees must work for Managed Bus stops	✓	Bharath Adithya, 6/21/2025, 10:40 AM
Phone_Number_Validation	Top of Page	Phone no must be 10 digits and starts with 6 or 7 or 8 or 9	✓	Bharath Adithya, 6/21/2025, 10:42 AM

## Milestone 8: Flow Setup

- 
- Designed a **record-triggered flow**:
    - Automatically fetches fare based on route + model.
  - Improved automation and reduced manual input errors.
- 

### OUTPUT SCREENSHOT



## Milestone 9: Apex Trigger

- Developed a trigger to validate:
    - Only users with role “Driver” can be added as Driver.
    - Only “Conductor” can be added as Conductor.
  - Ensured business logic is enforced at data level.

## OUTPUT SCREENSHOT

The screenshot shows the Salesforce Apex code editor with the following details:

- Title Bar:** The URL is `orgfarm-ed7725a0e4-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`.
- File Menu:** File, Edit, Debug, Test, Workspace, Help.
- Code Coverage:** None.
- API Version:** 64.
- Code:** The code is a class named `TripTriggerHandlerClass` with a static method `driverValidation`. The code checks if a driver ID exists in the database and adds an error to the trip record if it does not.

```
1 + Public class TripTriggerHandlerClass {
2
3 // Checking whether the entered Driver Id belongs to a Driver or not
4
5 + public Static void driverValidation(List<Trip__c> tripList){
6
7     List<Employee__c> driverList = [SELECT Id, Name FROM Employee__c WHERE Role__c ='Driver' ];
8
9     If(driverList != null){
10
11         Map<Id, String> driverMap = new Map<Id, String>();
12
13         for(Employee__c emp : driverList ){
14
15             driverMap.put(emp.Id, emp.Name);
16
17         }
18
19         for(Trip__c trip : tripList ){
20
21             If(trip.Driver_Id__c!=null){
22
23                 Boolean hasDriverId = driverMap.containsKey(trip.Driver_Id__c); // hasDriverId will be true
24
25                 If(hasDriverId == false){
26
27                     trip.addError('The assigned person is not a Driver.');
28
29             }
30
31         }
32
33     }
34
35 }
```

The screenshot shows the Salesforce Apex code editor with the file `TripTriggerHandlerClass.apex` open. The code implements a validation rule for the `Trip__c` object. It queries a list of conductors and creates a map to check if the entered conductor ID exists. If it does not, an error is added to the trip record.

```
38 // Checking whether the entered conductor Id belongs to a Conductor or not
39
40 public Static void conductorValidation(List<Trip__c> triplist){
41
42     List<Employee__c> conductorList = [SELECT Id, Name FROM Employee__c WHERE Role__c ='Conductor' ];
43
44
45
46     Map<Id, String> conductorMap = new Map<Id, String>();
47
48     for(Employee__c emp : conductorList ){
49
50         conductorMap.put(emp.Id, emp.Name);
51
52     }
53
54
55     for(Trip__c trip : triplist ){
56
57         If(trip.Conductor__c!=null){
58
59             Boolean hasConductorId = conductorMap.containsKey(trip.Conductor__c); // hasConductorId will be true
60
61             If(hasConductorId == false){
62
63                 tripaddError('The assigned person is not a Conductor.');
64
65             }
66
67         }
68
69     }
70
71
72 }
```

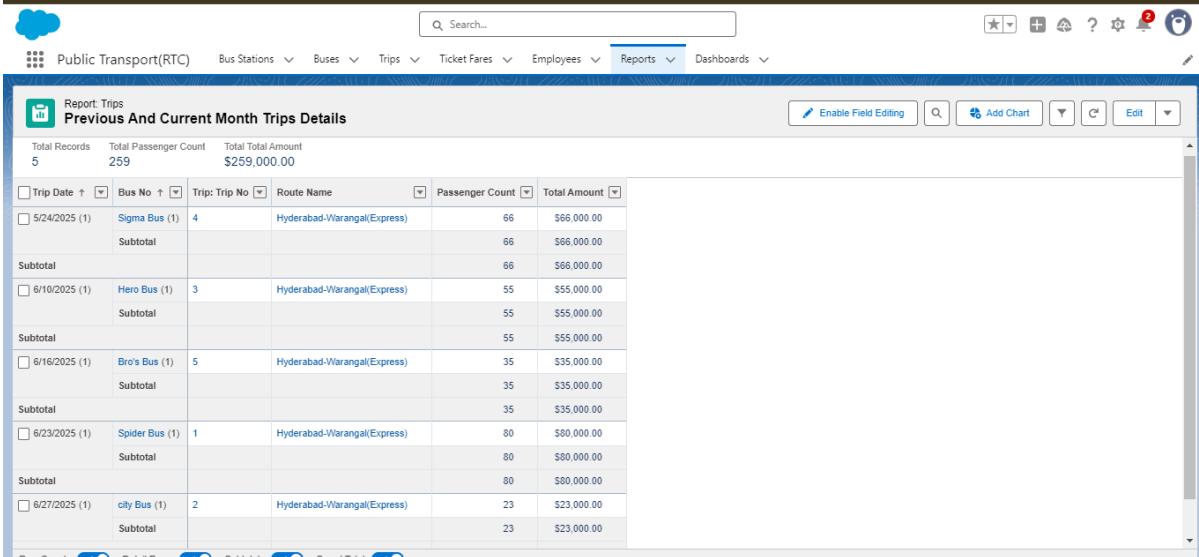
The screenshot shows the Salesforce Apex code editor with the file `TripTrigger.apex` open. It defines a trigger on the `Trip__c` object that runs before insert or update. The trigger logic calls methods from `TripTriggerHandlerClass` to validate the conductor ID and the trip ID.

```
1 trigger TripTrigger on Trip__c (before insert, before update) {
2
3
4
5     if(trigger.isBefore){
6
7         if(trigger.isInsert || trigger.isUpdate){
8
9             // Validating the Conductor Id in Trip is really a Conductor or not
10
11             TripTriggerHandlerClass.driverValidation(trigger.new);
12
13
14             // Validating the Conductor Id in Trip is really a Conductor or not
15
16             TripTriggerHandlerClass.conductorValidation(trigger.new);
17
18         }
19
20     }
21
22 }
```

## Milestone 10: Reports

- 
- Created reports for:
    - Employees per station, monthly revenue, trip counts.
  - Used grouping, filters, and field summaries.
  - Enabled data-driven decision-making.
- 

### OUTPUT SCREENSHOT



The screenshot shows a web-based reporting interface for Public Transport (RTC). The top navigation bar includes links for Bus Stations, Buses, Trips, Ticket Fares, Employees, Reports (which is currently selected), and Dashboards. The main content area displays a report titled "Report: Trips Previous And Current Month Trips Details". The report summary shows 5 total records, 259 total passenger count, and a total amount of \$259,000.00. The data is presented in a table with columns for Trip Date, Bus No, Trip No, Route Name, Passenger Count, and Total Amount. The table lists trips for five different buses across six dates. Each bus has a subtotal row, and the table ends with a grand total row. At the bottom of the report, there are checkboxes for Row Counts, Detail Rows, Subtotals, and Grand Total.

Total Records	Total Passenger Count	Total Total Amount
5	259	\$259,000.00
<input type="checkbox"/> Trip Date ↑ <input checked="" type="checkbox"/> Bus No ↑ <input type="checkbox"/> Trip: Trip No ↑ <input type="checkbox"/> Route Name ↑ <input type="checkbox"/> Passenger Count ↑ <input type="checkbox"/> Total Amount ↑		
□ 5/24/2025 (1)	Sigma Bus (1)	4
		Hyderabad-Warangal(Express)
		66
		\$66,000.00
	Subtotal	
		66
		\$66,000.00
	Subtotal	
□ 6/10/2025 (1)	Hero Bus (1)	3
		Hyderabad-Warangal(Express)
		55
		\$55,000.00
	Subtotal	
		55
		\$55,000.00
	Subtotal	
□ 6/16/2025 (1)	Bro's Bus (1)	5
		Hyderabad-Warangal(Express)
		35
		\$35,000.00
	Subtotal	
		35
		\$35,000.00
	Subtotal	
□ 6/23/2025 (1)	Spider Bus (1)	1
		Hyderabad-Warangal(Express)
		80
		\$80,000.00
	Subtotal	
		80
		\$80,000.00
	Subtotal	
□ 6/27/2025 (1)	city Bus (1)	2
		Hyderabad-Warangal(Express)
		23
		\$23,000.00
	Subtotal	
		23
		\$23,000.00
<input checked="" type="checkbox"/> Row Counts <input checked="" type="checkbox"/> Detail Rows <input checked="" type="checkbox"/> Subtotals <input checked="" type="checkbox"/> Grand Total		

Report: Employees  
**Employees By Bus Station**

Total Records 17

<input type="checkbox"/> Bus Station Name ↑	Employee: Employee Id	Employee Name	Role
<input type="checkbox"/> Central Bus Station (1)	1	Hero	Driver
Subtotal			
<input type="checkbox"/> Dwaraka Bus Station Complex (4)	2	Hemanth	Driver
	3	RK	Driver
	4	Bunny	Driver
	5	Tahir	Driver
Subtotal			
<input type="checkbox"/> Kakinada APSRTC Bus Station (5)	6	praveen	Driver
	7	Gangadhar	Driver
	8	sai	Driver
	17	maheesh	Conductor
	9	prem	Conductor
Subtotal			
<input type="checkbox"/> Simhachalam Bus Station (7)	10	Bharath	Conductor
	11	Bill	Conductor
	12	jathin	Conductor
	13	sindhu	Conductor
	14	Tammai	Conductor
	15	lalitha	Conductor
Row Counts	<input type="checkbox"/>	Detail Rows	<input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/>	Subtotals	<input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/>	Grand Total	<input checked="" type="checkbox"/>

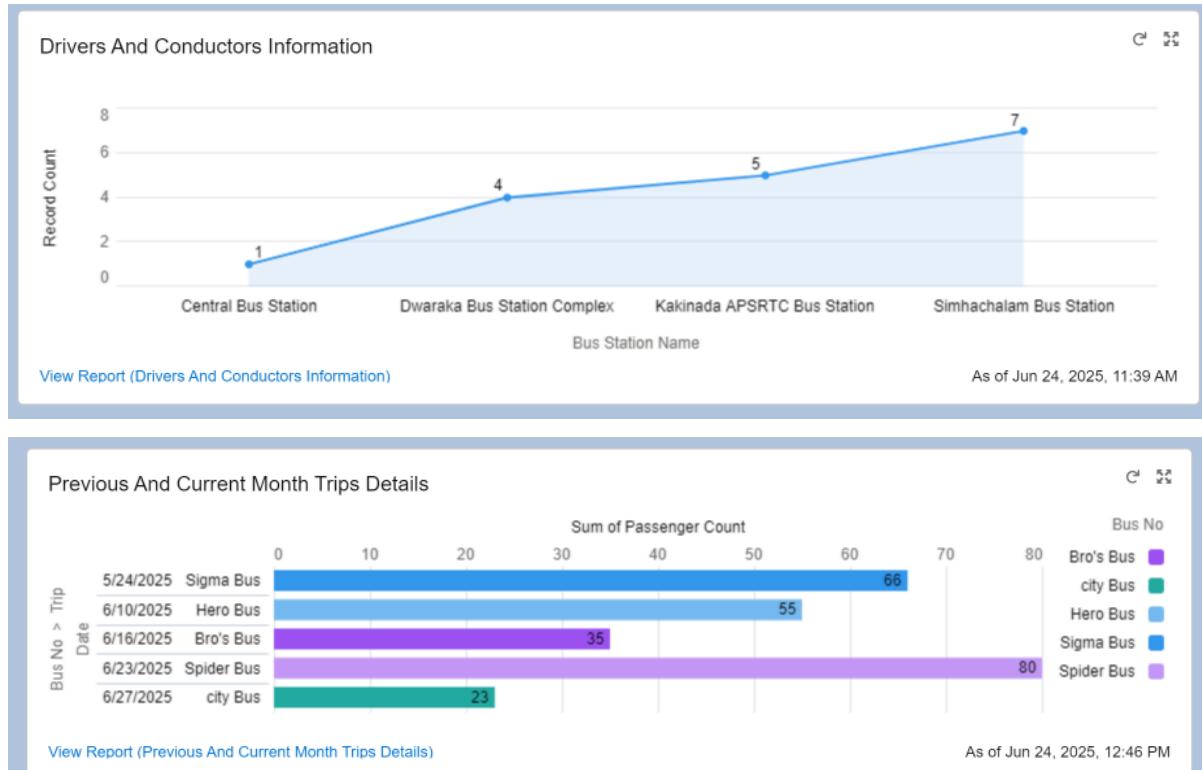
Report: Employees  
**Drivers And Conductors Information**

<input type="checkbox"/> Bus Station Name ↑	Employee: Employee Id	Employee: ID	Employee Name	Role
<input type="checkbox"/> Central Bus Station (1)	1	a04gL000005LOUH	Hero	Driver
Subtotal				
<input type="checkbox"/> Dwaraka Bus Station Complex (4)	2	a04gL000005LO7h	Hemanth	Driver
	3	a04gL000005LOCX	RK	Driver
	4	a04gL000005N8TX	Bunny	Driver
	5	a04gL000005NODF	Tahir	Driver
Subtotal				
<input type="checkbox"/> Kakinada APSRTC Bus Station (5)	6	a04gL000005LO65	praveen	Driver
	7	a04gL000005LOAv	Gangadhar	Driver
	8	a04gL000005LOE9	sai	Driver
	17	a04gL000005LOFI	maheesh	Conductor
	9	a04gL000005LOHN	prem	Conductor
Subtotal				
<input type="checkbox"/> Simhachalam Bus Station (7)	10	a04gL000005LNun	Bharath	Conductor
	11	a04gL000005LOlz	Bill	Conductor
	12	a04gL000005LOKb	jathin	Conductor
Row Counts	<input type="checkbox"/>	Detail Rows	<input checked="" type="checkbox"/>	Subtotals
	<input checked="" type="checkbox"/>	Grand Total	<input checked="" type="checkbox"/>	

## Milestone 11: Dashboards

- Designed dashboards showing:
  - Revenue trends, trip performance, and role summary.
- Used bar, pie, and summary widgets for visualization.

### OUTPUT SCREENSHOT



## Milestone 12: Final Review

---

- Conducted full system testing.
  - Checked all object links, flows, triggers, reports.
  - Confirmed the CRM is fully functional and stable.
- 

## 6.2 Dataset

---

The system was tested with the following types of data:(Sample Records and Input Values Used During Testing)

Object	Sample Fields Used
Bus Station	Name: Visakhapatnam Central, Category: Managed
Bus	Model: Super Deluxe, Capacity: 45, Station: Visakhapatnam
Employee	Name: Rakesh, Role: Driver, Phone: 9876543210, DOB: 1980-06-14
Ticket Fare	Route: Hyderabad-Warangal (Express), Fare: ₹220
Trip	Route: Hyderabad-Warangal, Passengers: 34, Estimated Time: 4 hrs

 Note: All test records were created using Salesforce's UI and validated via flows and formula fields

# 7. FUNCTIONAL AND PERFORMANCE TESTING

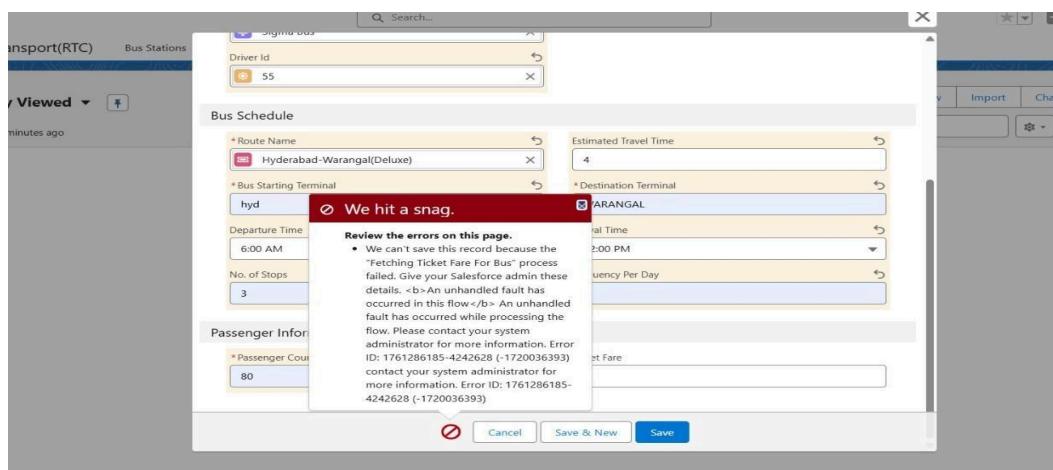
## 7.1 Performance Testing

The system was tested using realistic data entries.

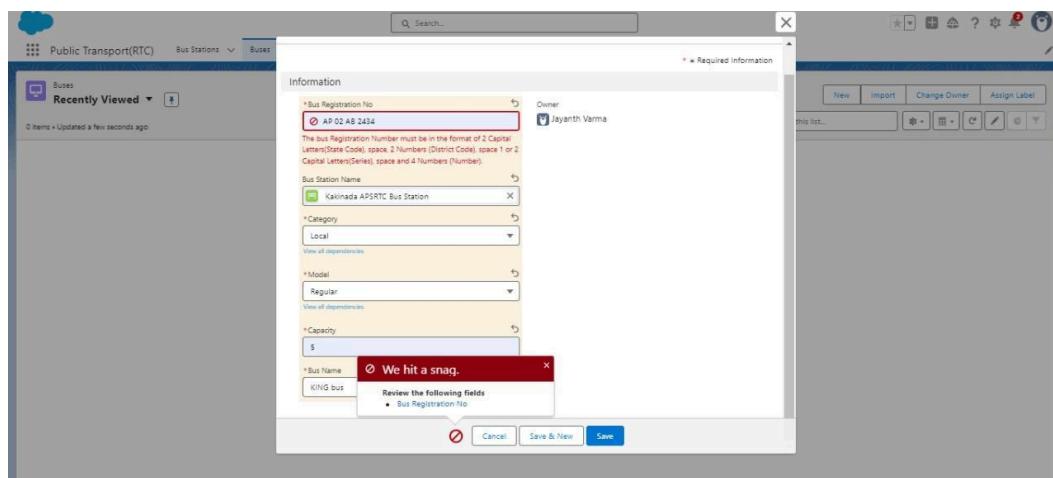
Key validations included:

- Trigger for role validation (Driver/Conductor)
- Flow for auto-fetching ticket fare
- Field validation for formats (e.g., registration number)  
Screenshots demonstrating these were captured during testing phases.

- Flow Automation Error (Fare Fetch Failure)



- Validation Rule Error (Bus Registration Format)



## 8. RESULTS

### 8.1 Output Screenshots

Key functional screenshots include:

- Correctly filled record (Trip, Fare, etc.)

The screenshot shows a software interface for managing public transport. At the top, there's a navigation bar with icons for cloud, search, Public Transport(RTC), Bus Stations, Buses, Trips, Ticket Fares (which is selected and highlighted in blue), Employees, and Reports. Below the navigation is a header bar with a ticket icon and the text "Ticket Fare" followed by "Hyderabad-Warangal(Express)". The main content area has tabs for "Related" and "Details". Under the "Details" tab, there are several fields: "Route Name" (Hyderabad-Warangal(Express)), "Owner" (Bharath Adithya with a profile icon), "Ticket Fare" (\$1,000.00), "Bus Model" (Regular), "Created By" (Bharath Adithya, 6/21/2025, 12:17 PM), and "Last Modified By" (Bharath Adithya, 6/23/2025, 5:39 AM). The background features a map-like pattern.

- Final report previews with grouped data

The screenshot shows a report titled "Report: Employees Drivers And Conductors Information". The top navigation bar includes icons for cloud, search, Public Transport(RTC), Bus Stations, Buses, Trips, Ticket Fares, Employees, Reports (selected and highlighted in blue), and Dashboards. There's also a "Enable Field Editing" button. The report itself displays a table with 17 total records. The columns are: Bus Station Name, Employee: Employee Id, Employee: ID, Employee Name, and Role. The data is grouped by bus station. For "Central Bus Station", there is one record for "Hero" as a Driver. For "Dwaraka Bus Station Complex", there are five records for "Hemanth", "RK", "Bunny", and "Tahir" as Drivers. For "Kakinada APSRTC Bus Station", there are three records for "praveen", "Gangadhar", and "sai" as Drivers, and one record for "mahesh" as a Conductor. Row counts, detail rows, subtotals, and grand total buttons are at the bottom of the table.

Bus Station Name	Employee: Employee Id	Employee: ID	Employee Name	Role
Central Bus Station (1)	1	a04gL000005LOUH	Hero	Driver
Subtotal				
Dwaraka Bus Station Complex (4)	2	a04gL000005LO7h	Hemanth	Driver
	3	a04gL000005LOCX	RK	Driver
	4	a04gL000005N8TX	Bunny	Driver
	5	a04gL000005NODF	Tahir	Driver
Subtotal				
Kakinada APSRTC Bus Station (5)	6	a04gL000005LO65	praveen	Driver
	7	a04gL000005LOAv	Gangadhar	Driver
	8	a04gL000005LOE9	sai	Driver
	17	a04gL000005LOFI	mahesh	Conductor

- Trigger error and fare auto-fill during Trip creation

New Trip

\* = Required Information

**Information**

*Trip No	35	Conductor Id	16
*Trip Date	6/28/2025	Owner	Bharath Adithya
*Bus No	Sigma Bus	Driver Id	3

Bus Schedule

\*Route Name

Cancel Save & New Save

- Validation error on form submission

We hit a snag.

Review the errors on this page.

- We can't save this record because the "Fetching Ticket Fare For Bus" process failed. Give your Salesforce admin these details. <b>An unhandled fault has occurred in this flow.</b> An unhandled fault has occurred while processing the flow. Please contact your system administrator for more information. Error ID: 1761286185-4242628 (-1720036393) contact your system administrator for more information. Error ID: 1761286185-4242628 (-1720036393)

Cancel Save & New Save

---

## **9. ADVANTAGES & DISADVANTAGES**

### **Advantages**

- Real-time data validation and automation
- Modular structure allows easy customization
- Centralized and streamlined process for trip, fare, and employee management
- Visual reports and dashboards for decision-making

### **Disadvantages**

- Requires knowledge of Salesforce for advanced use
  - Limited functionality in offline mode due to cloud dependency
  - Dependent on correctly established object relationships
- 

## **10. CONCLUSION**

The project successfully delivered a tailored Salesforce CRM application for RTC operations. It addressed key inefficiencies and brought structure to transport management workflows. The solution also served as a valuable learning experience for the team in both declarative and programmatic Salesforce development.

---

## 11. FUTURE SCOPE

The CRM can be extended further with features such as:

- Third-party ticketing and payment system integration
  - Automated alerts and communication features
  - Mobile-first user interface with Salesforce Experience Cloud
  - AI-powered forecasting using Salesforce Einstein
- 

## 12. APPENDIX

- Source Code: Attach Apex Trigger, Flow logic files



A screenshot of the Salesforce Apex code editor. The URL in the address bar is `orgfarm-ed7725a0e4-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab title is `TripTriggerHandlerClass.apc`. The code editor displays the following Apex trigger code:

```
trigger TripTrigger on Trip__c (before insert, before update) {
    if(trigger.isBefore){
        if(trigger.isInsert || trigger.isUpdate){
            // Validating the Conductor Id in Trip is really a Conductor or not
            TripTriggerHandlerClass.driverValidation(trigger.new);

            // Validating the Conductor Id in Trip is really a Conductor or not
            TripTriggerHandlerClass.conductorValidation(trigger.new);
        }
    }
}
```

- GitHub Link:  
<https://github.com/bharathadithya03/A-CRM-Application-For-Public-Transport-Management-System>
-

**Thank You**