Project: PromoPilot – Campaign Performance Management System for Retail Chains

1. Introduction

This document outlines the Low-Level Design (LLD) for **PromoPilot**, a campaign performance management system designed for retail chains to plan, execute, and analyze promotional campaigns across multiple outlets. The system enables campaign scheduling, budget tracking, customer engagement analysis, and ROI reporting.

Supports backend development using **Java (Spring Boot)** and **.NET (ASP.NET Core)**.

2. Module Overview

2.1 Campaign Planning & Scheduling Module
2.2 Budget Allocation & Spend Tracking Module
2.3 Store-Level Execution & Feedback Module
2.4 Customer Engagement & Redemption Tracking Module
2.5 ROI Analytics & Performance Reporting Module

3. Architecture Overview

3.1 Architectural Style

- **Frontend**: Angular/React for campaign dashboards and store portals.
- **Backend**: REST APIs for campaign orchestration and analytics.
- **Database**: SQL Server/PostgreSQL for structured campaign and transaction data.

3.2 Component Interaction

- Frontend allows campaign creation and monitoring.
- Backend processes campaign data and generates performance insights.

4. Module-Wise Design

4.1 Campaign Planning & Scheduling Module

4.1.1 Features

- Create campaigns with start/end dates, target products, and regions.
- Schedule campaigns across multiple stores.

4.1.2 Entities

- **Campaign**
    - CampaignID
    - Name
    - StartDate
    - EndDate
    - TargetProducts
    - StoreList

4.2 Budget Allocation & Spend Tracking Module

4.2.1 Features

- Allocate budgets per store or region.
- Track actual spend vs. planned budget.

4.2.2 Entities

- **Budget**
    - BudgetID
    - CampaignID
    - StoreID
    - AllocatedAmount
    - SpentAmount

4.3 Store-Level Execution & Feedback Module

4.3.1 Features

- Track campaign execution status at store level.
- Collect feedback from store managers.

4.3.2 Entities

- **ExecutionStatus**
    - StatusID
    - CampaignID
    - StoreID
    - Status
    - Feedback

4.4 Customer Engagement & Redemption Tracking Module

4.4.1 Features

- Track coupon redemptions, product purchases during campaigns.
- Analyze customer participation and repeat visits.

4.4.2 Entities

- **Engagement**
  - EngagementID
  - CampaignID
  - CustomerID
  - RedemptionCount
  - PurchaseValue

4.5 ROI Analytics & Performance Reporting Module

4.5.1 Features

- Calculate ROI, campaign reach, and conversion rates.
- Generate comparative reports across regions.

4.5.2 Entities

- **CampaignReport**
  - ReportID
  - CampaignID
  - ROI
  - Reach
  - ConversionRate
  - GeneratedDate

5. Deployment Strategy

5.1 Local Deployment

- Developer machines with mock campaign data.

5.2 Staging and Production Environments

- Cloud deployment with centralized campaign control.

6. Database Design

6.1 Tables and Relationships

- Campaign → Budget → ExecutionStatus → Engagement → CampaignReport

7. User Interface Design

7.1 Wireframes

- Campaign Dashboard: Active campaigns, budget status.
- Store Portal: Execution checklist and feedback form.
- Analytics Console: ROI graphs and regional comparisons.

8. Non-Functional Requirements

8.1 Performance

- Support 2,000 concurrent users across stores and HQ.

8.2 Usability

- Easy-to-use campaign builder and visual budget tracker.
- Multilingual support for regional stores.

8.3 Security

- Role-based access for marketing, finance, and store teams.
- Secure campaign data and audit logs.

8.4 Scalability

- Support for hundreds of campaigns across thousands of stores.

9. Assumptions and Constraints

9.1 Assumptions

- Stores will report campaign execution status daily.
- Customers will engage via coupons and loyalty programs.

9.2 Constraints

- Initial rollout for seasonal and product-specific campaigns only.