P.O BOX 63
Buea, Cameroon
Phone No:
(237)233322134/
233322690
Fax: (237)2 33322272
Email: info@ubuea.cm

REPUBLIC OF CAMEROON
PEACE-WORK-FATHERLAND

MINISTER OF HIGHER EDUCATION

UNIVERSITY OF BUEA

DEPARTMENT OF COMPUTER ENGINEERING

Course Code: CEF440

Course name - Internet Programming And Mobile Programming

Course Instructor - Dr Nkemeni Valery

System Modelling And Design

Task 4

Project 1

Task Report

Prepared by :

GROUP 6

Prepared on:
Sunday 26th May 2025

Group Members

NAME	MATRICULE
AJIM NEVILLE BEMSIBOM	FE22A141
EBUA CINDY SANGHA	FE22A195
KINGO KINGSLEY KAAH	FE22A233
NGONG ODILO BERTILA DUFE	FE22A263
NGOUNOU NGNINKEU JOEL JONATHAN	FE22A265

Table of Content

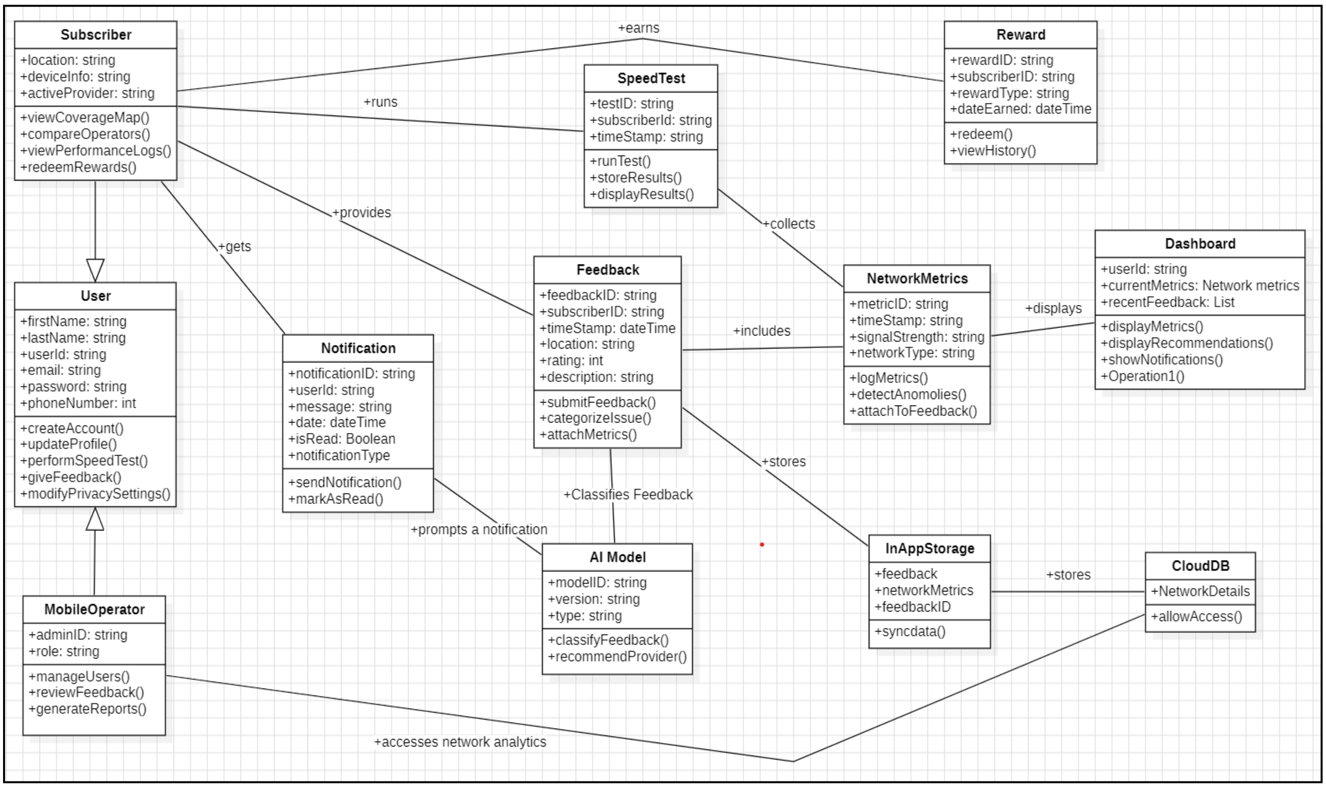
INTRODUCTION

This document presents a comprehensive analysis and visualization of the system architecture and behavior through a series of diagrams that serve as essential tools for understanding and communicating system requirements and functionality.

The following key diagrams are included in this document:

- **Context Diagram**: Offers a high-level view of the system, identifying external entities and their interactions with the system.
- **Dataflow Diagram (DFD)**: Illustrates how data moves within the system, highlighting processes, data stores, and data flows.
- **Use Case Diagram**: Describes the functional requirements of the system from the user's perspective, showing the interactions between actors and use cases.
- **Sequence Diagram**: Details the sequence of interactions between system components and users over time for specific scenarios.
- **Class Diagram**: Represents the static structure of the system, showing classes, attributes, methods, and relationships among objects.

Class Diagram



System Architecture

The class diagram represents a well-structured mobile application that operates with both local storage and cloud synchronization capabilities. The system follows object-oriented design principles with clear separation of concerns and modular architecture to ensure maintainability and scalability.

Core Classes Description

1. User Class

The User class serves as the foundation for user management within the system. It handles user authentication, profile management, and core user operations.

Purpose: Manages user accounts, authentication, and basic user operations including speed tests and feedback submission.

2. Subscriber Class

The Subscriber class extends the User class and represents active network subscribers who participate in the data collection process.

Purpose: Represents mobile network subscribers with additional functionality for network analysis and reward redemption.

3. NetworkMetrics Class

The NetworkMetrics class is responsible for collecting and managing all network-related performance data.

Purpose: Captures, stores, and processes objective network performance measurements automatically in the background.

4. Feedback Class

The Feedback class manages the collection and processing of subjective user experience data.

Purpose: Handles user feedback collection, categorization, and storage with associated network metrics.

5. SpeedTest Class

The SpeedTest class provides users with tools to manually test their network performance.

Purpose: Enables users to perform on-demand network speed tests and view results.

6. AI Model Class

The AI Model class represents the artificial intelligence components that provide advanced analytics and predictions.

Purpose: Provides intelligent analysis of network data and user feedback through machine learning models.

7. Notification Class

The Notification class manages all system notifications and user prompts.

Purpose: Handles intelligent notification delivery to users based on network conditions and user context.

8. Reward Class

The Reward class implements the incentive system to encourage user participation.

Purpose: Manages the reward system that incentivizes users to provide consistent feedback.

9. Dashboard Class

The Dashboard class provides users with a comprehensive view of their network experience and system information.

Purpose: Displays real-time network metrics, recent feedback, and recommendations in an intuitive interface.

10.InAppStorage Class

Manages local data storage for offline functionality.

Purpose: Provides local data storage capabilities for offline operation and data caching.

11.CloudDB Class

Handles cloud-based data storage and synchronization.

Purpose: Manages secure cloud storage and data synchronization across devices.

12.MobileOperator Class

The MobileOperator class represents network service providers and their administrative functions.

Purpose: Provides network operators with access to aggregated analytics and user feedback data.

Class Relationships and Interactions

Inheritance Relationships

Parent Class	Child Class	Relationship Type	Description
User	Subscriber	Inheritance	Subscriber extends User with additional network-specific functionality
User	MobileOperator	Inheritance	MobileOperator extends User with administrative capabilities

Association Relationships

Class A	Class B	Relationship	Multiplicity	Description
Subscriber	SpeedTest	runs	1:*	One subscriber can run multiple speed tests
Subscriber	Reward	earns	1:*	One subscriber can earn multiple rewards

User	Notification	gets	1:*	One user can receive multiple notifications
User	Feedback	provides	1:*	One user can provide multiple feedback entries
Feedback	NetworkMetrics	includes	1:1	Each feedback includes corresponding network metrics
NetworkMetrics	InAppStorage	stores	*:1	Multiple metrics stored in local storage
Feedback	AI Model	classifies	*:1	AI model classifies multiple feedback entries
Notification	AI Model	prompts	1:1	AI model determines when to send notifications
InAppStorage	CloudDB	stores	1:1	Local storage syncs with cloud database
MobileOperator	CloudDB	accesses	1:1	Mobile operators access cloud analytics
Dashboard	NetworkMetrics	displays	1:*	Dashboard displays multiple network metrics

Composition Relationships

Container Class	Component Class	Description
Feedback	NetworkMetrics	Feedback contains network metrics data
Dashboard	NetworkMetrics	Dashboard contains current metrics display
Subscriber	Dashboard	Subscriber has an integrated dashboard

Data Flow and System Operations

Primary Data Collection Flow

1. **Automatic Metrics Collection**: The NetworkMetrics class continuously collects network performance data in the background
2. **Anomaly Detection**: AI Model analyzes metrics to detect performance anomalies
3. **Intelligent Prompting**: Notification class prompts users for feedback when anomalies are detected
4. **Feedback Collection**: Feedback class collects user input and attaches corresponding network metrics
5. **Data Storage**: Both feedback and metrics are stored locally and synchronized with cloud storage
6. **Analysis and Recommendations**: AI Model processes data to provide network recommendations

User Interaction Flow

1. **User Registration**: User class handles account creation and authentication
2. **Background Operation**: System operates silently collecting metrics via NetworkMetrics class
3. **Speed Testing**: Users can initiate speed tests through SpeedTest class
4. **Feedback Submission**: Users provide feedback through Feedback class interface
5. **Reward Earning**: Reward class tracks participation and allocates incentives
6. **Dashboard Viewing**: Dashboard class displays real-time information and recommendations

System Integration Points

Mobile Platform Integration

Component	Platform Integration	Purpose
NetworkMetrics	Android WorkManager / iOS Background Fetch	Background data collection
User	Platform Authentication APIs	Secure user authentication

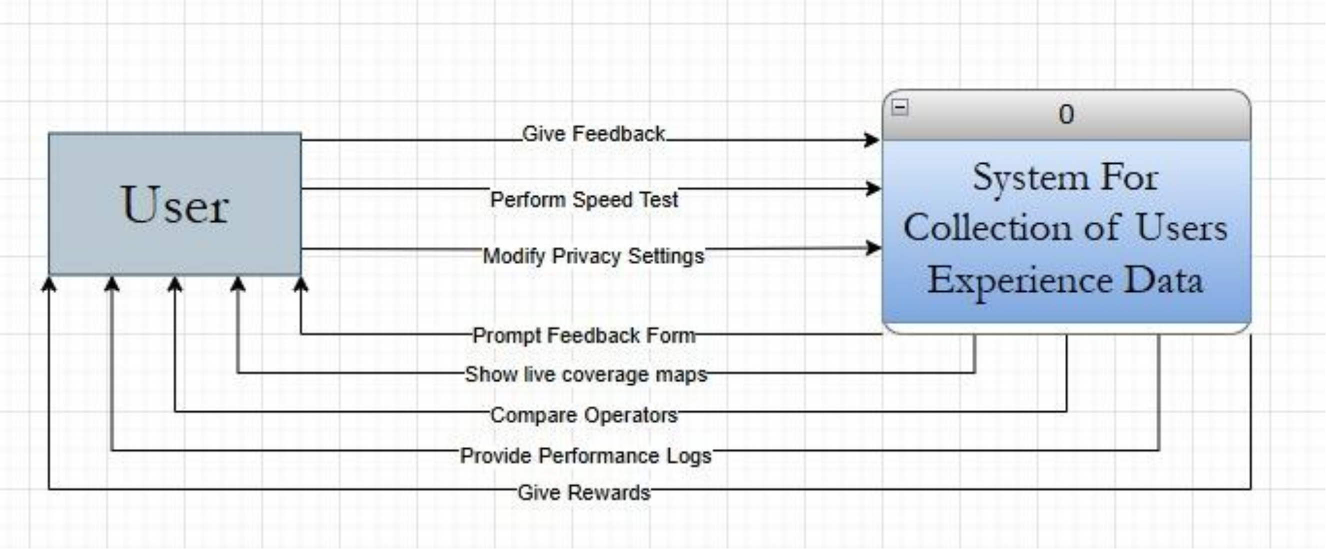
SpeedTest	Platform Network APIs	Network performance measurement
Notification	Platform Notification Services	User engagement

Cloud Services Integration

Component	Cloud Service	Purpose
CloudDB	PostgreSQL/MongoDB	Data storage and analytics
AI Model	TensorFlow Lite/Cloud ML	Machine learning inference
User	Firebase Authentication	User management
Reward	Payment Gateway APIs	Reward redemption

CONTEXT DIAGRAM

The diagram illustrates the interaction between the user and the system at a high level. The **User** is the external entity that communicates with the system through various actions such as giving feedback, performing speed tests, and modifying privacy settings. In response, the system provides functionalities including prompting feedback forms, showing live coverage maps, comparing operators, providing performance logs, and giving rewards.

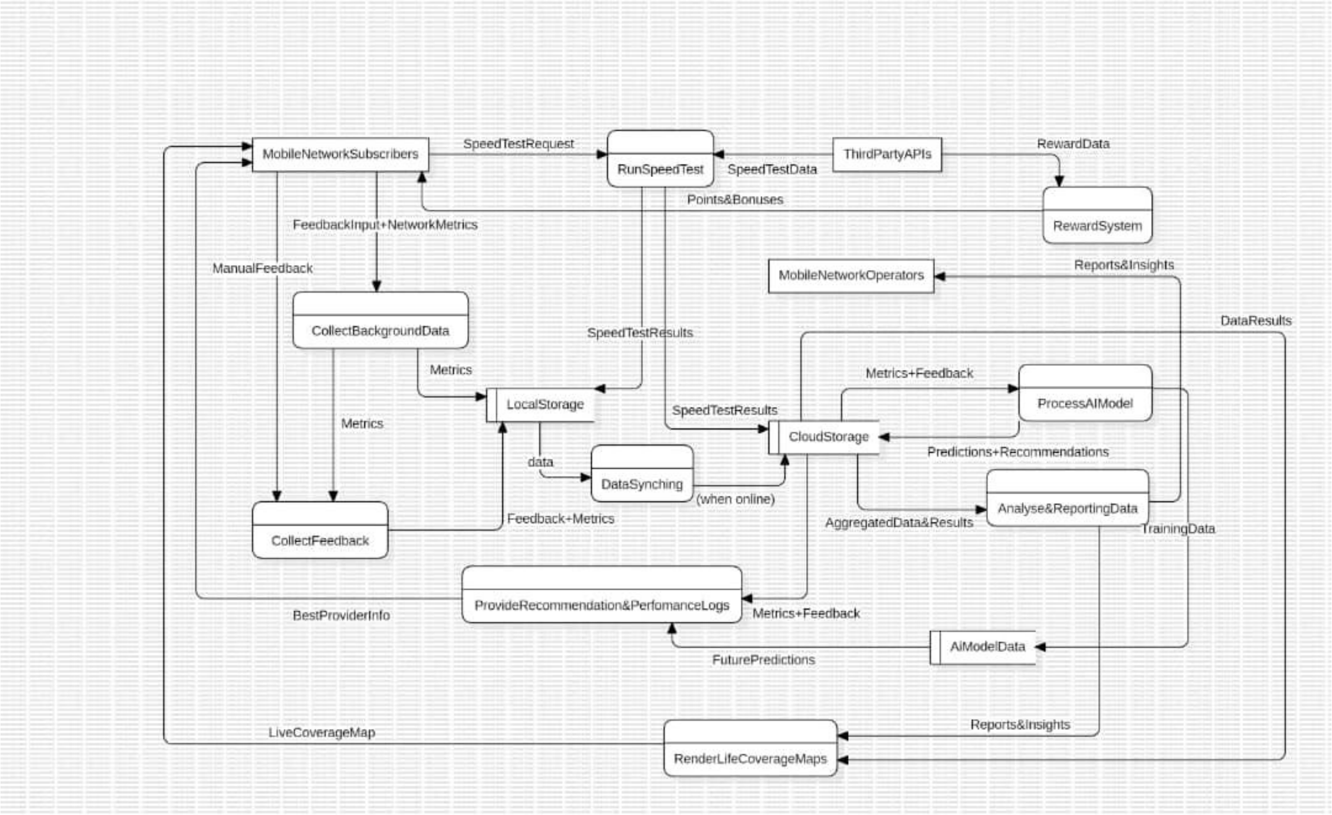


DATAFLOW DIAGRAM (DFD):

Below is a **Level-1 DFD** for the mobile application described in the Software Requirements Specification (SRS) document. A Level-1 DFD provides a high-level overview of the system, breaking down the main process into key subprocesses while showing interactions with external entities and data stores.

Components of the DFD

- **External Entities**: Mobile Network Subscribers, Mobile Network Operators, Third-Party APIs (e.g., Speedtest SDK, telecom reward APIs).
- **Processes**: Background Data Collection, Feedback Collection, Speed Test, Provider Recommendation, Reward System, Data Analysis & Reporting, AI Model Processing.
- **Data Stores**: Local Storage (for offline caching), Cloud Database (for synced data), AI Model Data (for training and predictions).
- **Data Flows**: Represent the movement of data (e.g., network metrics, user feedback, rewards) between entities, processes, and data stores.



1. **Mobile Network Subscribers**:

  - **Role**: Primary users who provide feedback (manual or prompted), initiate speed tests, and receive recommendations and rewards.
  - **Data Flows**:
    - Send **Network Metrics** (e.g., signal strength, latency, jitter) and **Feedback Input** (e.g., satisfaction ratings, issue descriptions) to the **Background Data Collection** and **Feedback Collection** processes.
    - Request **Speed Tests** and receive **Best Provider Info** and **Rewards** (points, airtime bonuses).

2. **Background Data Collection**:
   - **Function**: Automatically collects network metrics (e.g., signal strength, latency, network type, geolocation) in the background using APIs like Android's TelephonyManager (SRS Page 11).
   - **Data Flows**:
     - Stores metrics in **Local Storage (D1)** for offline caching.
     - Triggers **Feedback Collection** when metrics fall below thresholds (smart prompts, SRS Page 14).
     - Sends metrics to **Cloud Database (D2)** when online.
3. **Feedback Collection**:
   - **Function**: Handles both smart (triggered by poor metrics) and manual feedback submissions, auto-filling fields like timestamp, location, and network stats (SRS Page 24).
   - **Data Flows**:
     - Receives **Feedback Input** from subscribers and **Metrics** from Background Data Collection.
     - Stores feedback and metrics in **Local Storage (D1)** and syncs to **Cloud Database (D2)**.
4. **Speed Test**:
   - **Function**: Allows users to test upload/download speeds and ping using third-party libraries (e.g., Speedtest SDK, SRS Page 14).
   - **Data Flows**:

        - Receives **Speed Test Requests** from subscribers and **Speed Test Data** from Third-Party APIs.
        - Stores results in **Local Storage (D1)** and syncs to **Cloud Database (D2)**.

5. **Provider Recommendation**:
    - **Function**: Uses crowdsourced data to suggest the best network provider based on location and real-time metrics, with future AI enhancements (SRS Page 14).
    - **Data Flows**:
        - Pulls **Metrics + Feedback** from **Cloud Database (D2)**.
        - Outputs **Best Provider Info** to subscribers.
6. **Reward System**:
    - **Function**: Tracks user contributions (feedback, data sharing) and assigns points or airtime bonuses, integrated with telecom APIs (SRS Page 14).
    - **Data Flows**:
        - Receives **Reward Data** from Third-Party APIs.
        - Sends **Points, Bonuses** to subscribers.
7. **AI Model Processing**:
    - **Function**: Processes metrics and feedback to predict Quality of Experience (QoE) and categorize issues using models like XGBoost or DistilBERT (SRS Page 15).
    - **Data Flows**:
        - Pulls **Metrics + Feedback** from **Cloud Database (D2)**.
        - Stores **Predictions, Recommendations** in **AI Model Data (D3)** and syncs to **Cloud Database (D2)**.
8. **Data Analysis & Reporting**:
    - **Function**: Generates insights and reports for network operators, visualizing trends and user satisfaction (SRS Page 5).
    - **Data Flows**:

    - Pulls **Aggregated Data** from **Cloud Database (D2)**.
    - Sends **Reports, Insights** to **Mobile Network Operators**.

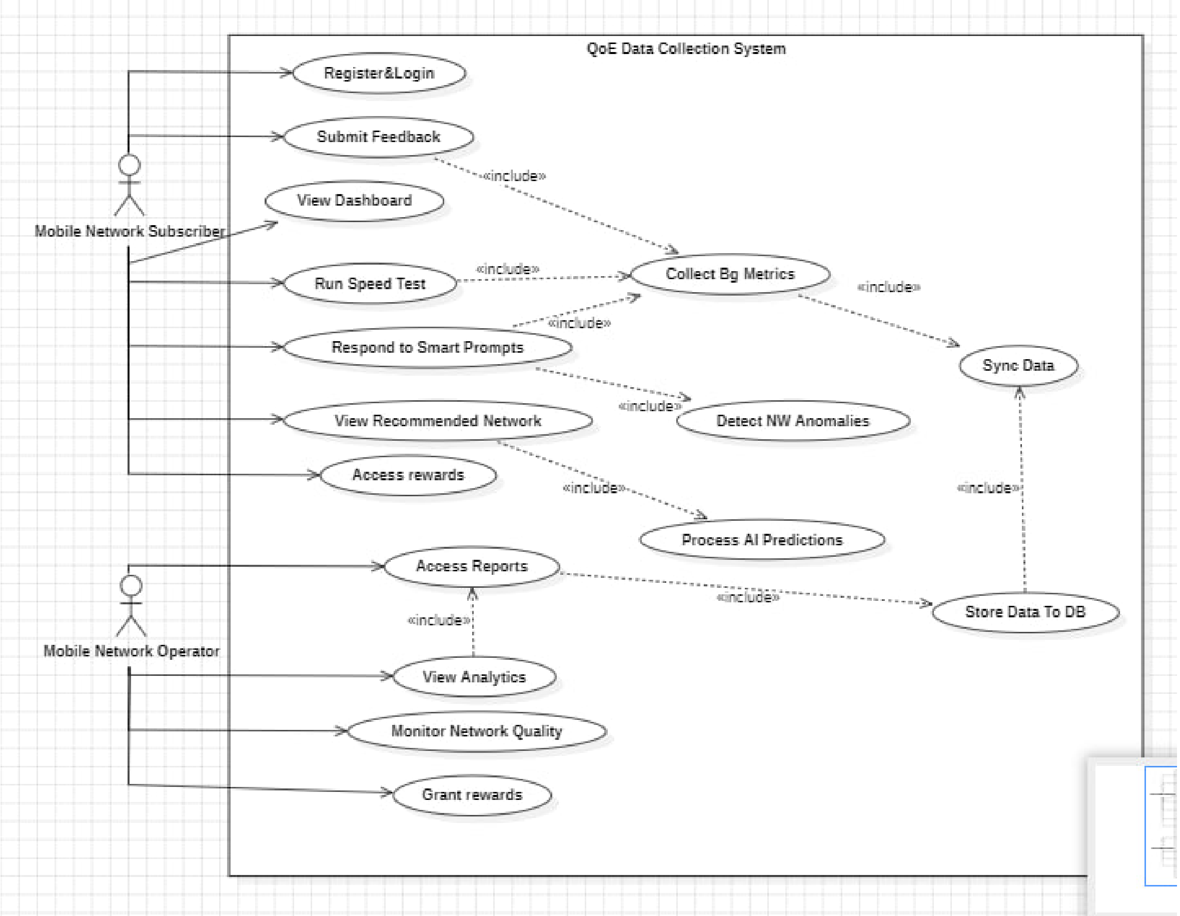
9. **Mobile Network Operators**:
    - **Role**: Use reports and insights to improve services, which indirectly benefits subscribers (SRS Page 6).
    - **Data Flows**:
        - Receive **Reports, Insights** from Data Analysis & Reporting.
        - Provide **Service Improvements** to subscribers.
10. **Data Stores**:
    - **Local Storage (D1)**: Caches metrics, feedback, and speed test results offline, syncing to the cloud when connectivity is restored (SRS Page 15).
    - **Cloud Database (D2)**: Stores all synced data (PostgreSQL/MongoDB) for analysis, recommendations, and reporting (SRS Page 16).
    - **AI Model Data (D3)**: Stores data for training and predictions, used in future phases for AI-driven features (SRS Page 25).

USE CASE DIAGRAM

This use case diagram illustrates the interactions between two main user types:

- Mobile Network Subscribers
- Mobile Network Operators

It also describes the functionalities of a QoE (Quality of Experience) Data Collection System. The system aims to collect, process, and analyze network performance and user feedback to enhance the overall quality of mobile services.



Key Actors

1. Mobile Network Subscriber

End users of mobile services who interact with the system to provide feedback, view reports, and receive network-related recommendations and rewards.

2. Mobile Network Operator

Service providers who use the system to monitor network quality, analyze performance metrics, and grant user incentives based on insights.

Primary Use Cases for Subscribers

i. ***Register & Login***: Access the system securely.
ii. ***Submit Feedback:*** Share user experience with the network.
iii. ***View Dashboard:*** Visualize network performance metrics.
iv. ***Run Speed Test (includes Collect Bg Metrics):*** Measure real-time network speed.
v. ***Respond to Smart Prompts (includes Collect Bg Metrics):*** Answer context-aware questions for deeper analysis.
vi. ***View Recommended Network (includes Detect NW Anomalies):*** Get suggestions for optimal networks based on data.
vii. ***Access Rewards (includes Process AI Predictions*)**:** Earn incentives for participating in data collection.

Primary Use Cases for Operators

i. ***Access Reports (includes Store Data To DB):*** View comprehensive reports.
ii. ***View Analytics (includes Access Reports):*** Analyze trends and metrics.
iii. ***Monitor Network Quality*:** Observe performance across locations and times.Grant Rewards: Distribute rewards to participating users.

System-Level Use Cases

i. ***Collect Bg Metrics (background metrics):*** Core function to gather performance data.
ii. ***Sync Data (includes Collect Bg Metrics):*** Upload metrics to the central server.
iii. ***Detect NW Anomalies:*** Identify abnormal patterns in network performance.
iv. ***Process AI Predictions***: Utilize AI to interpret data and predict network issues.
v. ***Store Data To DB*:** Persistently save collected and processed data.

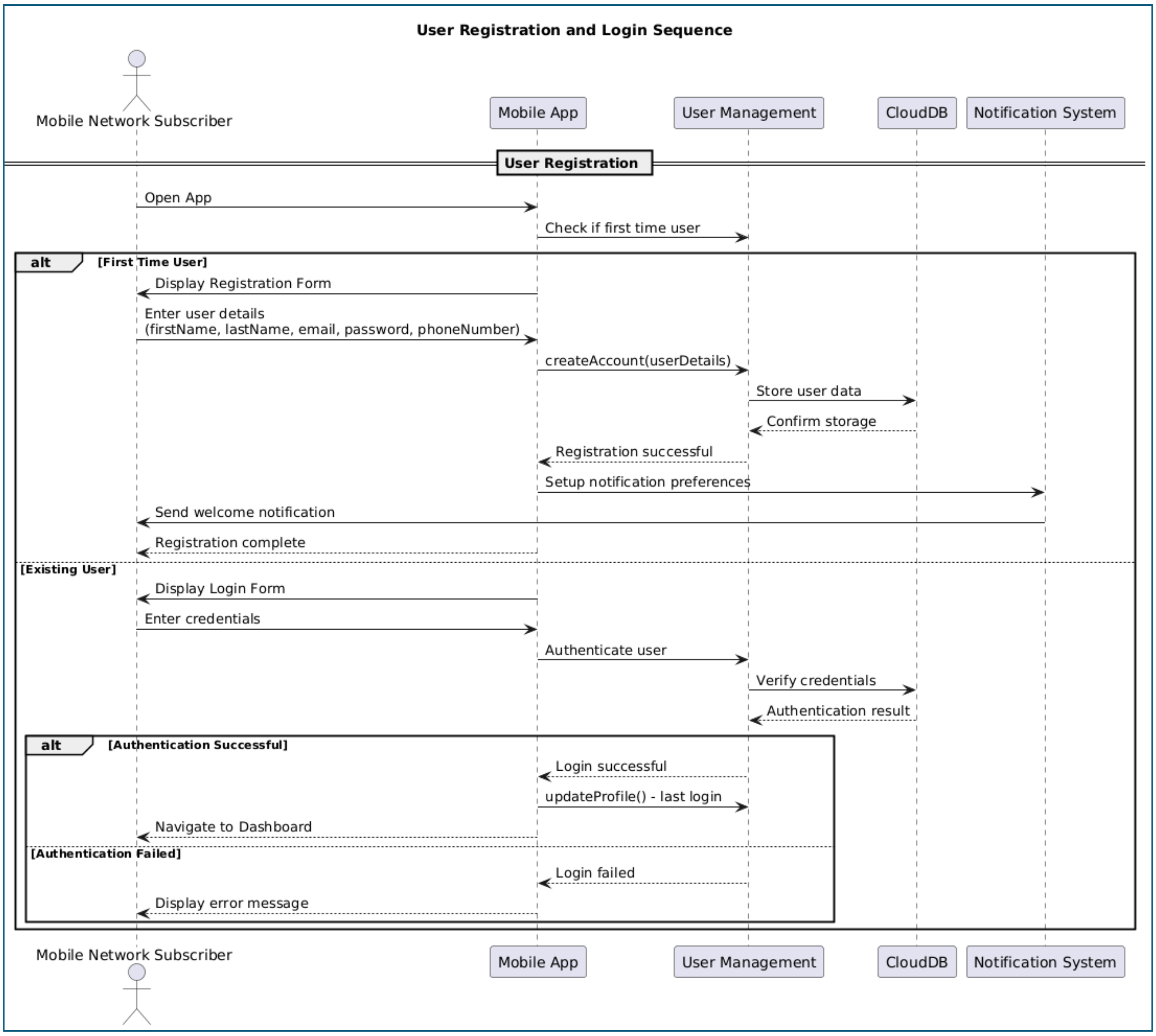
Relationships

Include Relationships are heavily used to denote that certain use cases (e.g., Run Speed Test, Submit Feedback) rely on core system functionalities like Collect Bg Metrics, Sync Data, and Store Data to DB.

SEQUENCE DIAGRAM

1. User Registration and Login Sequence

This diagram illustrates sequence which handles both new user registration and existing user authentication processes



New User Registration Process

For first-time users, our system presents a comprehensive registration form that captures essential subscriber information. The registration process involves several critical steps:

Data Collection Phase: Our application prompts users to enter their personal details including first name, last name, email address, password, and phone number. This information is crucial for creating personalized user profiles and enabling targeted feedback collection.

Account Creation: Once the user submits their information, our User Management component processes the registration request by invoking the createAccount() method. This triggers a series of backend operations where our system stores the user data securely in our CloudDB infrastructure.

Notification Setup: Following successful account creation, our system automatically configures notification preferences for the new user. This setup is essential for our smart prompt system, which will later deliver timely feedback requests based on network performance patterns.

Welcome Process: Our Notification System sends a personalized welcome notification to the newly registered user, confirming successful registration and introducing them to our data collection methodology.

Existing User Authentication

For returning users, our system streamlines the login process while maintaining security protocols:

Credential Verification: Users enter their login credentials, which our User Management component validates against our secure database. Our CloudDB performs the authentication verification and returns the result to the management layer.

Successful Authentication Path: When authentication succeeds, our system updates the user's profile with the current login timestamp and navigates them directly to the main dashboard, where they can immediately access all system features.

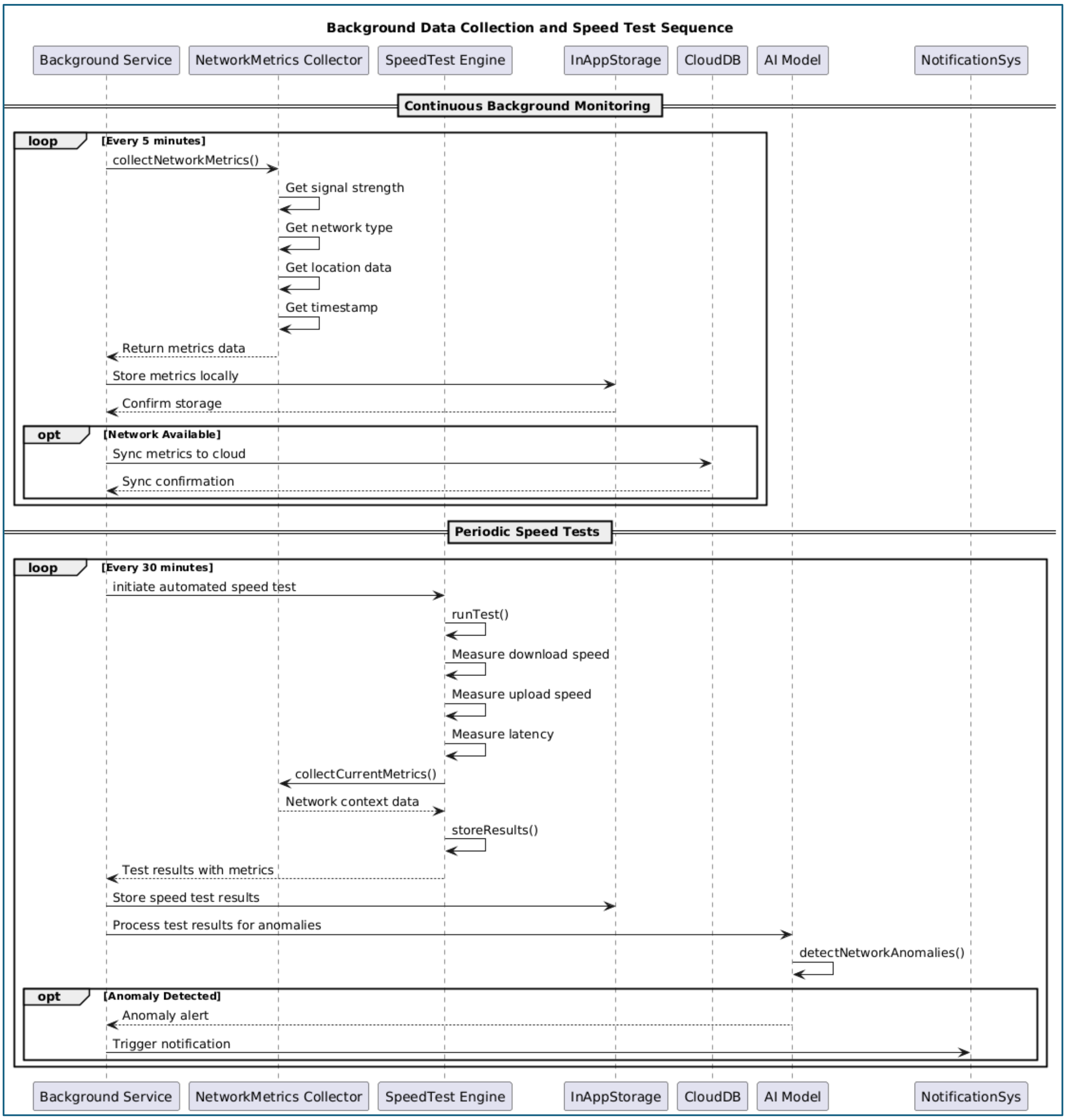
Failed Authentication Handling: In cases where authentication fails, our system provides clear error messaging and allows users to retry the login process, ensuring a user-friendly experience while maintaining security.

Integration Points

This sequence establishes critical integration points with our notification system and database infrastructure, creating the necessary foundation for all subsequent user interactions within our QoE data collection ecosystem.

2. Background Data Collection and Speed Test Sequence

Our second sequence diagram represents the core functionality of our QoE data collection system. This sequence operates continuously in the background, automatically gathering network performance metrics and conducting periodic speed tests without requiring user intervention.



Continuous Background Monitoring

Our background service operates on a precisely timed schedule, executing data collection routines every five minutes to ensure comprehensive network performance monitoring:

Metrics Collection Cycle: Our NetworkMetrics Collector component systematically gathers critical network parameters including signal strength measurements, current network type identification, precise location data, and timestamp information. This multi-dimensional data collection approach ensures we capture the complete context of each user's network experience.

Local Storage Strategy: Immediately after collection, our system stores all metrics locally using our InAppStorage component. This local-first approach ensures data preservation even when network connectivity is intermittent, which is particularly important in developing countries where network reliability may vary.

Cloud Synchronization: When network connectivity is available, our system automatically synchronizes locally stored metrics with our CloudDB infrastructure. This optional synchronization process ensures data persistence while accommodating varying network conditions.

Periodic Speed Test Implementation

Our system conducts comprehensive speed tests every thirty minutes, providing objective performance measurements:

Automated Test Initiation: Our Background Service triggers speed tests automatically without user intervention, ensuring consistent data collection regardless of user activity or awareness.

Comprehensive Performance Measurement: Our SpeedTest Engine executes multiple performance assessments including download speed measurement, upload speed evaluation, and latency calculation. Each test provides a complete picture of current network performance.

Contextual Data Integration: During each speed test, our system simultaneously collects current network metrics to provide contextual information. This correlation between speed test results and environmental factors enables more accurate performance analysis.

Results Processing and Storage: Following each test, our system stores comprehensive results locally and processes them through our AI Model for immediate anomaly detection.

Intelligent Anomaly Detection

Our AI-powered analysis system continuously monitors performance data:

Pattern Recognition: Our AI Model analyzes speed test results and network metrics to identify performance anomalies and unusual patterns that may indicate network issues.

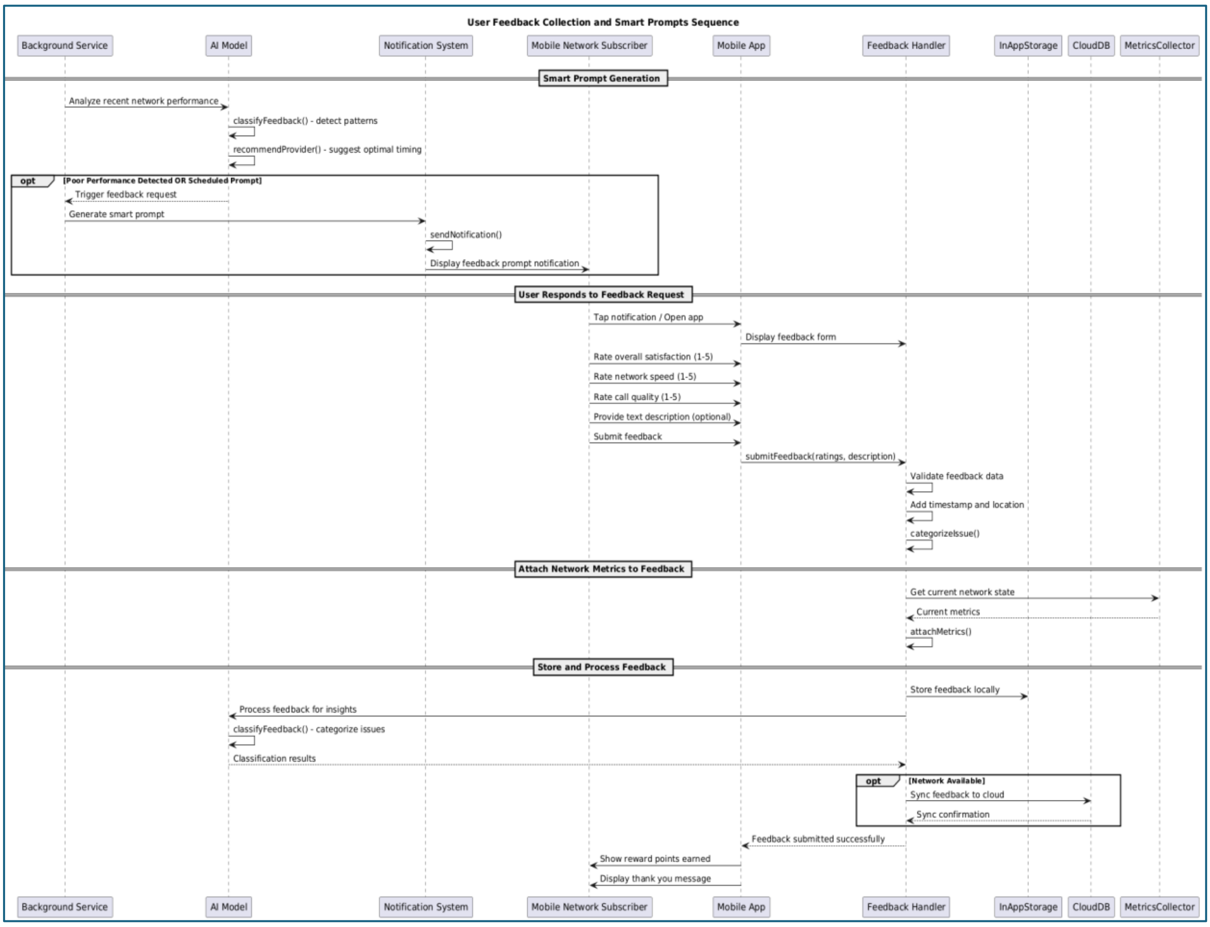
Proactive Alerting: When our system detects significant anomalies, it automatically triggers notifications to alert users about potential network issues, enabling proactive response to performance problems.

System Integration

This sequence demonstrates the seamless integration between our data collection components, storage systems, and intelligent analysis capabilities, creating a comprehensive background monitoring solution that operates transparently to users while providing valuable insights to network operators.

3. User Feedback Collection and Smart Prompts Sequence

Our third sequence diagram illustrates our intelligent feedback collection system, which combines AI-driven prompt generation with user-friendly feedback submission processes. This sequence represents the subjective data collection component of our QoE system.


User Feedback Collection and Smart Prompts Sequence
Background Service
AI Model
Notification System
Mobile Network Subscriber
Mobile App
Feedback Handler
InAppStorage
CloudDB
MetricsCollector
Smart Prompt Generation
Analyze recent network performance
classifyFeedback() - detect patterns
recommendProvider() - suggest optimal timing
opt
[Poor Performance Detected OR Scheduled Prompt]
Trigger feedback request
Generate smart prompt
sendNotification()
Display feedback prompt notification
User Responds to Feedback Request
Tap notification / Open app
Display feedback form
Rate overall satisfaction (1-5)
Rate network speed (1-5)
Rate call quality (1-5)
Provide text description (optional)
Submit feedback
submitFeedback(ratings, description)
Validate feedback data
Add timestamp and location
categorizeIssue()
Attach Network Metrics to Feedback
Get current network state
Current metrics
attachMetrics()
Store and Process Feedback
Store feedback locally
Process feedback for insights
classifyFeedback() - categorize issues
Classification results
opt
[Network Available]
Sync feedback to cloud
Sync confirmation
Feedback submitted successfully
Show reward points earned
Display thank you message
Background Service
AI Model
Notification System
Mobile Network Subscriber
Mobile App
Feedback Handler
InAppStorage
CloudDB
MetricsCollector

AI-Driven Smart Prompt Generation

Our system employs sophisticated artificial intelligence to optimize feedback collection timing and relevance:

Performance Analysis: Our AI Model continuously analyzes recent network performance data, identifying patterns and trends that indicate optimal moments for feedback collection. This analysis includes performance degradation detection, usage pattern recognition, and quality threshold monitoring.

Intelligent Classification: Through our classifyFeedback() method, our AI system categorizes performance patterns and identifies situations where user feedback would be most valuable. This classification helps prioritize feedback requests based on potential impact and relevance.

Optimal Timing Recommendations: Our recommendProvider() function suggests the most appropriate timing for feedback requests, considering factors such as recent network performance, user activity patterns, and historical response rates.

Smart Notification Delivery

When our AI determines that feedback collection would be valuable, our system initiates a carefully orchestrated notification process:

Trigger Conditions: Our system generates feedback requests either when poor performance is detected or according to scheduled prompt intervals designed to maintain consistent data collection.

Personalized Notifications: Our Notification System creates contextually relevant prompts that reference current network conditions, making feedback requests more meaningful and likely to generate responses.

User Feedback Submission Process

Our user interface design prioritizes simplicity and comprehensiveness in feedback collection:

Multi-Dimensional Rating System: Users provide ratings across multiple dimensions including overall satisfaction, network speed perception, and call quality assessment. Each rating uses a standardized 1-5 scale for consistency and ease of analysis.

Optional Descriptive Feedback: Our system accommodates users who wish to provide detailed descriptions of their experiences, capturing qualitative insights that complement quantitative ratings.

Automated Data Enhancement: Upon feedback submission, our system automatically enhances user-provided data with timestamp information, precise location data, and current network context.

Network Context Integration

Our system's strength lies in correlating subjective feedback with objective network measurements:

Real-Time Metrics Attachment: When users submit feedback, our system immediately captures current network metrics through our MetricsCollector component, ensuring perfect temporal alignment between subjective experiences and objective measurements.

Comprehensive Data Correlation: Our attachMetrics() function creates detailed associations between user perceptions and actual network performance, enabling sophisticated analysis of the relationship between objective performance and subjective experience.

Feedback Processing and Storage

Our backend systems process feedback through multiple layers of analysis:

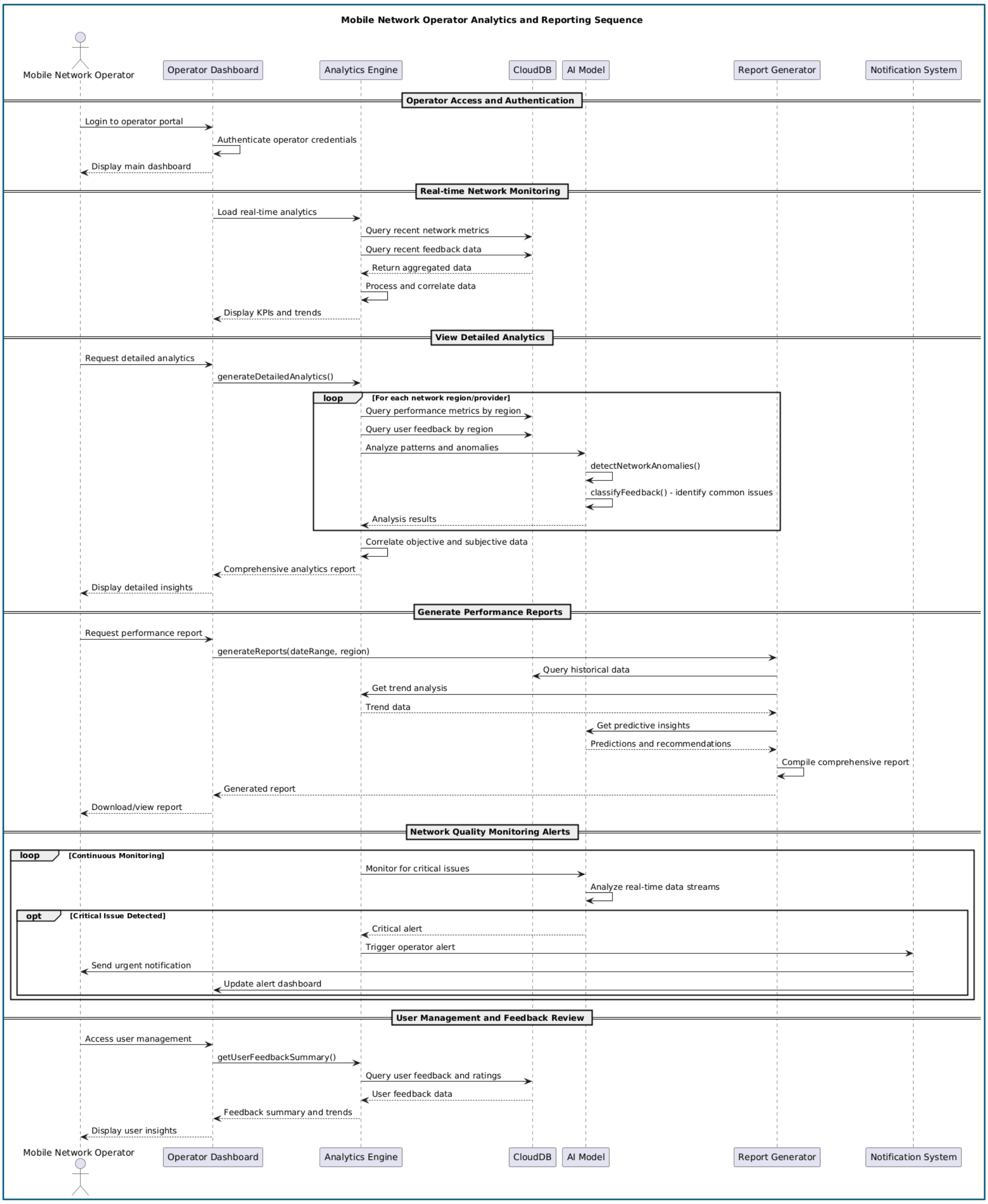
Local Storage First: All feedback is immediately stored locally to ensure data preservation, then synchronized with our cloud infrastructure when connectivity permits.

AI-Powered Classification: Our AI Model processes each feedback submission through our classifyFeedback() method, automatically categorizing issues and identifying common themes across user reports.

User Engagement: Following successful submission, our system rewards users with points and displays appreciation messages, encouraging continued participation in our data collection program.

4. Mobile Network Operator Analytics and Reporting Sequence

Our fourth and final sequence diagram demonstrates the comprehensive analytics and reporting capabilities we provide to mobile network operators. This sequence transforms collected data into actionable insights for network optimization and customer experience management.


Mobile Network Operator Analytics and Reporting Sequence
Mobile Network Operator
Operator Dashboard
Analytics Engine
CloudDB
AI Model
Report Generator
Notification System
Operator Access and Authentication
Login to operator portal
Authenticate operator credentials
Display main dashboard
Real-time Network Monitoring
Load real-time analytics
Query recent network metrics
Query recent feedback data
Return aggregated data
Process and correlate data
Display KPIs and trends
View Detailed Analytics
Request detailed analytics
generateDetailedAnalytics()
loop
[For each network region/provider]
Query performance metrics by region
Query user feedback by region
Analyze patterns and anomalies
detectNetworkAnomalies()
classifyFeedback() - identify common issues
Analysis results
Correlate objective and subjective data
Comprehensive analytics report
Display detailed insights
Generate Performance Reports
Request performance report
generateReports(dateRange, region)
Query historical data
Get trend analysis
Trend data
Get predictive insights
Predictions and recommendations
Compile comprehensive report
Generated report
Download/view report
Network Quality Monitoring Alerts
loop
[Continuous Monitoring]
Monitor for critical issues
Analyze real-time data streams
opt
[Critical Issue Detected]
Critical alert
Trigger operator alert
Send urgent notification
Update alert dashboard
User Management and Feedback Review
Access user management
getUserFeedbackSummary()
Query user feedback and ratings
User feedback data
Feedback summary and trends
Display user insights
Mobile Network Operator
Operator Dashboard
Analytics Engine
CloudDB
AI Model
Report Generator
Notification System

Operator Authentication and Access Control

Our system implements robust security measures for operator access:

Secure Portal Access: Mobile network operators access our analytics through a dedicated dashboard that implements comprehensive authentication protocols, ensuring that sensitive network performance data remains secure.

Role-Based Access: Our authentication system verifies operator credentials and provides appropriate access levels based on organizational roles and responsibilities.

Real-Time Network Monitoring Dashboard

Our analytics engine provides operators with immediate visibility into network performance:

Live Data Integration: Our Analytics Engine continuously queries our CloudDB for the most recent network metrics and user feedback data, presenting operators with real-time insights into network performance across their coverage areas.

Key Performance Indicators: Our dashboard displays critical KPIs including average network speeds, user satisfaction ratings, problem frequency distributions, and trend analyses that enable quick assessment of network health.

Data Correlation: Our system processes and correlates objective network measurements with subjective user feedback, providing operators with a comprehensive view of actual user experiences rather than just technical metrics.

Comprehensive Regional Analysis

Our analytics capabilities extend beyond basic monitoring to provide detailed regional insights:

Geographic Performance Mapping: Our system analyzes performance metrics and user feedback by geographic region, enabling operators to identify area-specific issues and optimization opportunities.

Provider Comparison: When applicable, our analytics engine compares performance across different network providers, helping operators understand their competitive position and identify improvement areas.

AI-Powered Pattern Detection: Our AI Model continuously analyzes data streams using our detectNetworkAnomalies() and classifyFeedback() methods, identifying patterns that might not be immediately apparent to human analysts.

Advanced Reporting Capabilities

Our reporting system transforms raw data into actionable business intelligence:

Customizable Report Generation: Operators can generate comprehensive reports for specific date ranges and geographic regions through our ReportGen component, which compiles data from multiple sources into cohesive analytical documents.

Trend Analysis Integration: Our reports include sophisticated trend analysis that helps operators understand performance trajectories and predict future network behavior.

Predictive Insights: Our AI Model provides predictive insights and recommendations based on historical data patterns, enabling proactive network management rather than reactive problem-solving.

Proactive Alert System

Our monitoring system includes sophisticated alerting capabilities:

Continuous Monitoring: Our Analytics Engine works in conjunction with our AI Model to continuously monitor data streams for critical issues that require immediate attention.

Intelligent Alert Generation: When critical issues are detected, our system automatically generates alerts through our Notification System, ensuring that operators receive timely notifications about urgent network problems.

Multi-Channel Notifications: Critical alerts are delivered through multiple channels including direct notifications to operators and updates to the alert dashboard, ensuring that important issues receive appropriate attention.

User Management and Feedback Analysis

Our system provides operators with comprehensive user experience insights:

Feedback Aggregation: Our analytics engine processes user feedback data to provide summaries and trend analyses that help operators understand customer satisfaction patterns and identify common concerns.

User Journey Analysis: Operators can analyze user feedback patterns over time, understanding how user experiences change in response to network modifications or external factors.

Actionable Insights: All analytics and reports focus on providing actionable insights that operators can use to make informed decisions about network investments, optimization priorities, and customer experience improvements.

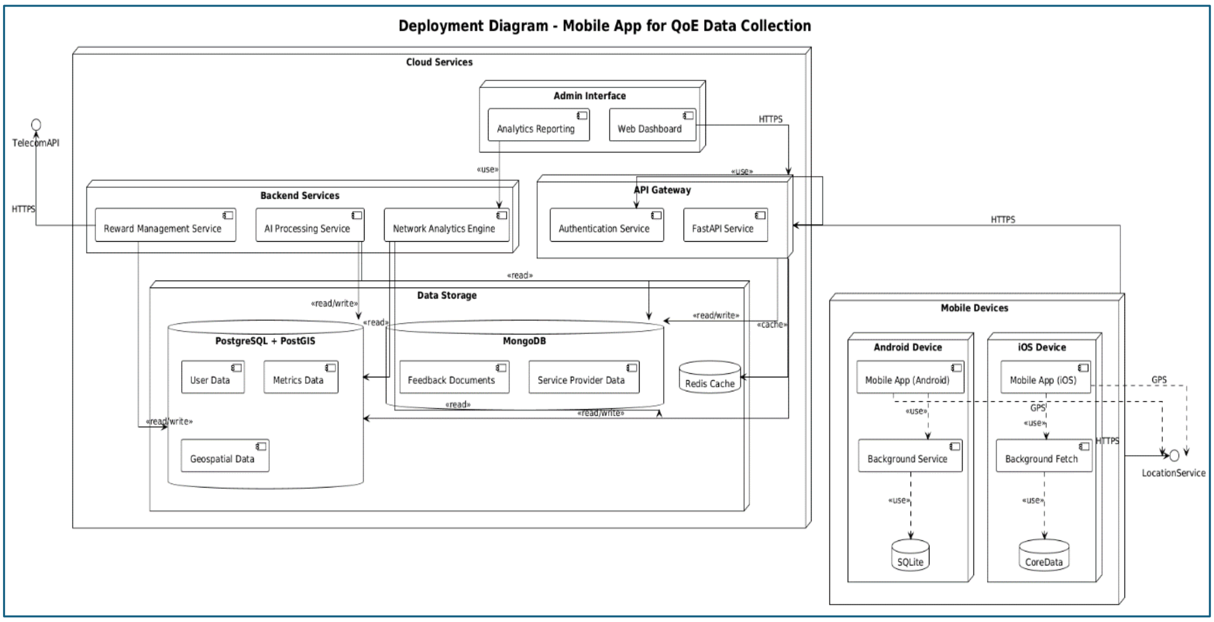
System Integration and Data Flow

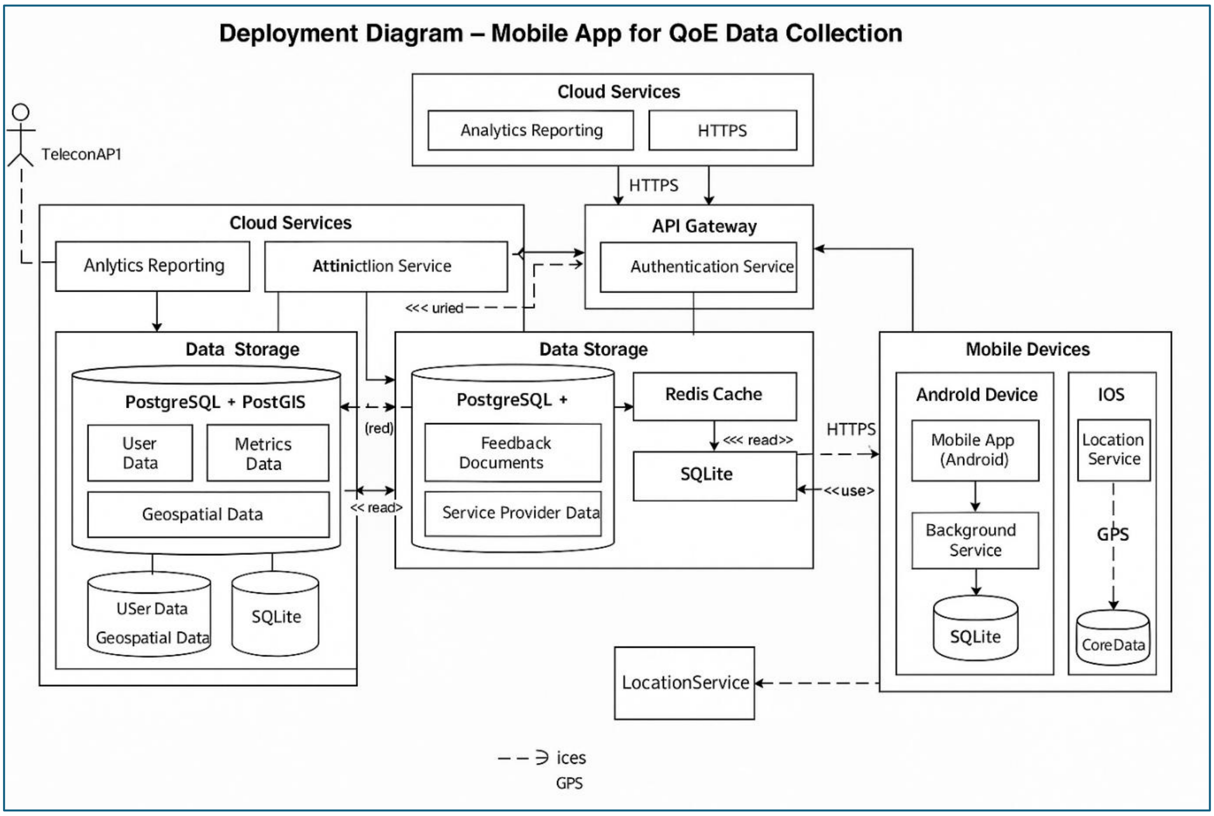
This sequence demonstrates the complete data flow from collection through analysis to actionable insights, showcasing how our comprehensive QoE system transforms raw user experience data into valuable business intelligence that enables mobile network operators to enhance service quality and customer satisfaction.

Our analytics and reporting capabilities represent the culmination of our data collection efforts, transforming the continuous stream of user feedback and network metrics into the insights necessary for evidence-based network management and optimization decisions.

Deployment Diagram

This diagram shows how our mobile app for collecting network quality data is set up and works. The system is built to do everything we need based on our project requirements, making sure users have a good experience while we collect useful data about mobile networks.





Our Mobile Apps - Running in the Background

Our mobile app works on both Android and iPhone devices. On Android, we use WorkManager to run tasks in the background without bothering users. On iPhone, we use Background Fetch to do the same thing. Both apps save data locally using SQLite (Android) and CoreData (iPhone), so the app still works even when there's no internet connection.

Our apps automatically collect important network information like what type of network you're on (3G, 4G, 5G), signal strength, how fast data loads, GPS location, time stamps, and which mobile company you're using. All of this happens quietly in the background, using very little battery power (less than 2% per day).

Our Cloud Services - The Smart Processing Part

Our backend services handle all the smart features we planned. The Network Analytics Engine looks at all the network data to spot problems using AI models like Isolation Forest. Our AI Processing Service predicts how good your network experience will be using models like XGBoost or neural networks.

We can run AI models both on the phone (using TensorFlow Lite) for quick responses and in the cloud for more complex processing. This gives us the best of both worlds - fast responses and powerful analysis.

Our Data Storage - Keeping Everything Organized

We use different types of databases for different jobs. PostgreSQL with PostGIS helps us work with location data for our network recommendations. MongoDB stores user feedback and comments because it's flexible. Redis Cache makes everything faster by storing frequently used data. This setup handles thousands of users at once and makes sure our app responds quickly when users want network recommendations for their area.

Our Security - Keeping Data Safe

All data moving between our app and servers uses HTTPS with strong encryption. Our Authentication Service uses JWT tokens to make sure only authorized users can access the system. This keeps user data safe and supports our privacy features. We can anonymize user data and let users choose what information they want to share, following privacy laws like GDPR.

Our Recommendation System - Finding the Best Networks

Our Location Services integration helps our recommendation engine work properly. We collect data from many users to tell others which mobile network works best in their area right now. This gives users real-time advice about which network to use. Our connection to Telecom APIs helps us detect which mobile company users are on and potentially gives out airtime rewards.

Our Scalable Setup - Growing with More Users

Our system is built with separate services that can grow independently. The API Gateway makes it easy to add new features and AI models as we improve the app. Everything runs in the cloud with auto-scaling, so we can handle more users automatically. We can deploy our system on different cloud providers and meet local regulations in different countries.

Our Management Tools - Monitoring Everything

Our Web Dashboard and Analytics Reporting let us see how the system is performing, how engaged users are, and how well our rewards system works. This helps us make good decisions and keep users happy while making sure we don't bother them too much. We sync data in batches to save bandwidth while making sure no data gets lost.

Our Core Features - What Users Actually Use

Our speed test feature works through the mobile apps, giving users a simple one-tap way to check their network speed. Our feedback system automatically fills in location, time, and network details, so users only need to describe their problem. This setup creates a solid foundation for our network quality app that's both technically advanced and easy for users to use, while following all the rules and keeping the app lightweight and efficient.

CONCLUSION

In conclusion, the report outlines a structured mobile application architecture that effectively collects and analyzes network performance metrics and user feedback. By employing object-oriented design principles, the system ensures maintainability and scalability.