

P.O BOX 63 Buea
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
ENGINEER

Course Code: CEF440

Internet Programming And Mobile Programming

Course Instructor: Dr Nkemeni Valery

SOFTWARE REQUIREMENT SPECIFICATION (SRS) DOCUMENT

**Project Name: A MOBILE APP FOR COLLECTION OF USERS EXPERIENCE DATA
FROM MOBILE NETWORK SUBSCRIBERS.**

Date: 5th May 2025

Version: 1

REVISION HISTORY		
Version	Version description	Date completed
1	The document encompasses the system's intended functionality and behaviour, and constraints.	05/ 05/ 2023

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 Contents

1. INTRODUCTION.....	4
2. REVIEW AND ANALYSIS OF REQUIREMENTS.....	7
1. Requirements Elicitation Sources.....	7
2. Requirements Analysis Techniques.....	9
3. INCONSISTENCIES, AMBIGUITIES AND MISSING INFORMATION.....	10
➤ Inconsistencies	10
➤ Ambiguities	10
➤ Missing or Sparse Information	10
4. Requirements Classification	11
6. Summary of Requirement Analysis Outcomes.....	13
3. REQUIREMENTS.....	14
3.1 Functional Requirements:.....	14
3.2 Non – Functional Requirements:.....	15
Technical Requirements:.....	16
UI/UX Requirements	17
4. PROJECT CONSTRAINTS:.....	19
5. ESTIMATED COST.....	19
6. PROJECT CALENDAR	20
MVP (Minimum Viable Product).....	21
Phase 2 – Core Enhancements	22
Objective: Increase user engagement and data quality.....	22
Phase 3 – Full Version with AI & Personalization.....	22
Objective: Use data to deliver intelligent, location-aware, and incentive-driven features... 22	
MVP Clickable Prototype Wireframe Overview	23
AI Model Placeholder (For Future Phase)	25

1. INTRODUCTION

This document serves as a comprehensive guide that outlines the goals, scope, and constraints of **the Mobile App for Collection of Users Experience Data from Mobile Network Subscribers**, providing a clear and concise understanding of what is expected from its solution.

This document also provides a detailed description of the software requirements of the system; it defines the functional and non-functional requirements. It will serve as a foundation for us, the stakeholders, end users, and all parties involved in the project to ensure a common understanding of the software's functionality and behaviour. The purpose of this document is to establish a shared understanding of the project's objectives, enabling effective communication between us the development team, and the stakeholders.

Throughout this document, you will find a detailed analysis of the system's requirements, including the functional capabilities, performance expectations and constraints.

Additionally, this document will help to manage project scope by defining the boundaries of the software solution. By clearly documenting the requirements and constraints, it becomes easier to track and manage changes that may occur during the development process.

It is important to note that this SRS document is a living document that will evolve throughout the project's lifecycle. As new information becomes available or requirements change, this document will be updated accordingly to ensure that all stakeholders are kept well informed.

By following the guidelines and specifications outlined in this document, the development team can create a software solution that meets the needs and expectations of the stakeholders, delivering a high-quality product that fulfils its intended purpose.

1.1. Product Scope:

The application is designed to serve multiple purposes, focusing on the collection of user experience data from mobile network subscribers. Key components of the scope include:

- ❖ **Continuous Background Operation:** The application will run in the background of users' devices, enabling it to gather data without requiring constant user interaction. This feature is essential for capturing real-time experiences and metrics.
- ❖ **Periodic User Feedback Prompts:** Users will receive prompts at regular intervals to provide feedback on their network experience. This feedback will include satisfaction ratings and comments about specific service aspects.

- ❖ **Automatic Logging of Performance Metrics:** The application will automatically log key device and network parameters. These metrics will include:
 - Network type (e.g., 2G, 3G, 4G, 5G)
 - Signal strength (in dBm)
 - Latency (response time)
 - Jitter (variation in packet arrival time)
 - Packet loss rates
 - Bandwidth usage
 - Geolocation data (timestamped)
- ❖ **Integration of Subjective and Objective Data:** The application will combine both subjective user feedback (qualitative data) and objective performance metrics (quantitative data) to provide a comprehensive view of the Quality of Experience (QoE).
- ❖ **Data Visualization and Reporting:** The app will offer insights through data visualization, allowing users and network operators to understand performance trends and user satisfaction levels.

1.2. Product Value:

The mobile application delivers substantial value to various stakeholders by addressing critical pain points in mobile network service quality. The key aspects of its value proposition include:

- ❖ **Enhanced Customer Satisfaction:** By enabling users to provide real-time feedback, the application empowers subscribers to voice their concerns, ultimately leading to improved satisfaction with network services.
- ❖ **Data-Driven Decision Making for Operators:** Mobile network operators will gain access to a wealth of data that reflects user experiences. This information can inform strategic decisions regarding network upgrades, maintenance, and service enhancements.
- ❖ **Proactive Network Management:** The application facilitates a proactive approach to network management by identifying issues before they escalate. Continuous data collection allows operators to respond swiftly to service disruptions.

- ❖ **User Empowerment:** Subscribers will feel more engaged and valued as their feedback directly influences service quality. This empowerment can foster loyalty and trust in the network provider.
- ❖ **Competitive Advantage:** By adopting such technology, mobile network operators can differentiate themselves in a competitive market, showcasing their commitment to quality service and customer care.

1.3. Intended Audience:

The mobile application is designed with two genres of audiences in mind:

- ❖ **Mobile Network Subscribers:** These are the end-users who utilize mobile network services.
- ❖ **Mobile Network Operators:** Companies such as MTN, Orange, and CAMTEL that manage mobile networks. They will leverage the data collected to enhance service delivery, optimize network performance, and engage with customers more effectively.

1.4. Intended Use:

The system is to be abstracted to the genres of users so s/he can perform what s/he intends to perform.

- ❖ **Mobile Network Subscribers:**
 - **Real-Time Feedback:** Users can provide immediate feedback regarding their network experience, facilitating a more responsive service environment.
 - **Issue Reporting:** The application will allow users to report issues like slow speeds or outages, ensuring that their concerns are documented and addressed.
- ❖ **Mobile Network Operators:**
 - **Data Analysis:** Operators will analyse the collected data to understand user behaviour, identify common issues, and make informed decisions about network improvements.
 - **User Engagement:** The feedback mechanism will enable operators to engage with users directly, enhancing customer relationships through targeted communication and solutions.

1.5. General Description:

The project focuses on the design and implementation of a mobile application aimed at collecting user experience data from mobile network subscribers. This initiative stems from

the recognition that user expectations for service quality and performance have risen significantly, particularly in Urban areas, where fluctuations in network quality are common. The application is intended to operate unobtrusively in the background, continuously collecting relevant data while periodically engaging users for their feedback. By combining subjective user inputs with objective performance metrics, the application seeks to provide a holistic understanding of the Quality of Experience (QoE) from the subscribers' perspective.

2. REVIEW AND ANALYSIS OF REQUIREMENTS

The goal here was to analyze the user and stakeholder requirements gathered for the mobile application intended to collect user experience data from mobile network subscribers in Cameroon. This analysis will inform the creation of the final Software Requirements Specification (SRS) document, guide implementation phases, and ensure a user-centric, technically feasible, and efficient system design.

1. Requirements Elicitation Sources

Requirements were gathered through:

- **User interviews and feedback sessions**, including responses captured in the `UserData.csv` file.
- **Observational research** on common complaints and needs of mobile subscribers.
- **Competitor analysis** (Reverse Engineering) of existing applications such as OpenSignal, MySpeed, and nPerf.

Features obtained from these systems

Feature	Source App	Upgrade/Improvement for Your App
Crowdsourced Network Quality Data	Opensignal	Expand to include real-time data submission + user feedback. Show live coverage maps and compare operators based on user location.
Interactive Coverage Maps	Opensignal	Add AI-powered heatmaps to predict best network per area and time of day. Enable community comments or alerts on map regions.
Passive Data Collection (e.g., signal, latency, jitter, packet loss)	QoEHero	Simplify for general users, but maintain accuracy. Make this run in the background with low battery impact . Only prompt users if thresholds are breached.

User-Friendly Testing Tool	Speed	Opensignal, Speedtest	Allow users to test upload/download speeds. Visualize results clearly and add option to compare with national averages or peers nearby .
Gamification & Incentive System		WiFiScout	Introduce points, badges, or airtime/data bonuses for active feedback and passive data contributions. Incentives should be location-aware and progressive.
Manual Submission with Contextual Auto-Data	Feedback	Instabug	Build an easy-to-use “Report Issue” button that auto-attaches network info, location, device, and time. Let users add text and optionally record voice complaints.
Operator Recommendation System		Custom feature (Not fully implemented elsewhere)	Use AI to analyze user feedback + metrics across regions , then recommend the best operator in real-time for a given area. Could be a widget or map overlay.
Real-time Notification of Network Status		Opensignal, QoEHero	Notify users when network degrades or improves significantly. Users can opt-in to get alerts or silence them.
Detailed Metrics Logging	Technical	QoEHero	Store metrics like RSSI/RSRP, CQI, SINR — but only show simplified insights unless user opts for advanced view. Store all in database for analytics and model training.
Secure, Background Data Syncing		Instabug	When offline, store logs locally , and auto-sync when online. Encrypt data in transit and at rest.
Device and OS Detection		Instabug	Useful for correlating issues with device types. Add statistics dashboards for admins/devs to spot patterns.
Feedback + Metrics Pairing for Model Training		Custom Idea (yours)	Store user issues alongside timestamped metrics to train ML models to predict dissatisfaction from passive data. Use supervised learning to classify network health.

- **Stakeholder meetings** to determine business goals and feasibility.

2. Requirements Analysis Techniques

a. Completeness

To assess completeness, we examined whether the collected requirements address all major aspects of the system:

- ✓ Functional features (data logging, feedback, speed testing, etc.)
- ✓ Non-functional qualities (lightweight, low power usage, privacy-respecting)
- ✓ Incentive systems (user motivation through bonuses)

The coverage was found to be adequate for defining a Minimum Viable Product (MVP) and planning future expansions.

b. Clarity

Clarity was achieved by translating ambiguous user input into specific, measurable system behaviors. For example:

- ✓ Input: "App shouldn't disturb me all the time"
- ✓ Translated Requirement:
 - 🚩 The app will prompt for feedback **only** when performance drops below predefined thresholds.

c. Technical Feasibility

- ✓ **Feasible** using native or cross-platform mobile development frameworks (e.g., Flutter, React Native).
- ✓ **Passive network monitoring:** Requires OS-specific APIs (e.g., TelephonyManager for Android).
- ✓ **Location services and background operation:** Feasible but must address battery consumption and permissions.
- ✓ **Reward system:** Needs integration with APIs from telecom operators or third-party SMS/data APIs.
- ✓ **Data Storage:** Cloud-based backend (e.g., Firebase, Supabase, or custom REST API).

d. Dependency Analysis

Identified dependencies include:

- ✓ Feedback submission requires access to real-time metrics and location.
- ✓ AI-based recommendations depend on an accumulated dataset.
- ✓ Rewards are based on valid user actions (feedback, participation).

e. Prioritization

Using feedback frequency and business value, requirements were prioritized into MVP features and future enhancements.

3. INCONSISTENCIES, AMBIGUITIES AND MISSING INFORMATION

➤ Inconsistencies

1. Location Formatting

- Variants like:
 - “Buea Southwest”
 - “Buea SW REGION”
 - “Buea, South West Region”
- 📌 This will cause grouping issues in analysis.

2. Network Providers

- Some listed as “MTN;Orange;Camtel”, others just “MTN”.
- No standardized delimiter (some use ;, others a comma or space).
- 📌 Affects filtering and aggregation.

3. Response Case Sensitivity

- “Yes” vs “yes”, “No” vs “no”.
- Some fields contain leading/trailing spaces.

4. Satisfaction Responses

- “Not applicable” in a field intended for rating satisfaction.
- 📌 Suggests a need for conditional visibility in the form (“If yes, then rate...”).

➤ Ambiguities

1. “What would stop you from using this app?”

Responses include vague entries like:

- “If my privacy is expised” (likely meant “exposed”)
- “No improvement with network services over time” (not directly a blocker for app use)

2. “What do you mostly use the network for?”

- Only 13/68 responded; unclear if this was optional or skipped due to misunderstanding.
- Some entered vague or overlapping terms like “personal”, “everything”.

3. Concerns About the App

- “Data usage” could mean mobile data cost or app overconsumption.
- “Battery drain” is understandable, but no scale of concern is given.

➤ Missing or Sparse Information

1. **Low Response Count in Some Fields**
 - “Username”: Only 5 entries
 - “What do you mostly use the network for?": Only 13 filled
 - Suggests optional fields or unclear purpose.
2. **No Detailed Feature Prioritization**
 - “What features would encourage...” gives multiple selections but no indication of priority or importance.
3. **Lack of Contextual Metadata**
 - No device model, network type (e.g., 4G, 3G), or time-on-network data.
 - This limits technical troubleshooting possibilities.
4. **No Geo-coordinates or structured region tagging**
 - Town/city input is free text, making geographical analysis harder.

4. Requirements Classification

A. Functional Requirements

- ✓ Real-time collection of network metrics (signal strength, latency, jitter, bandwidth).
- ✓ Smart triggering of user feedback forms based on metric thresholds.
- ✓ Manual feedback submission option.
- ✓ Speed test with downloadable/uploadable performance logs.
- ✓ Recommendation of best service providers based on location data.
- ✓ Reward point system based on feedback participation.

B. Non-Functional Requirements

- ✓ Lightweight design (<20MB APK size).
- ✓ Background operation without significant battery drain.
- ✓ Privacy-compliant storage of user data (GDPR-like behavior).
- ✓ Reliable data syncing when connectivity is restored.

C. Technical Requirements

- ✓ Use of Android TelephonyManager APIs for metrics.
- ✓ Integration with Google Play Services for location.
- ✓ Firebase or Supabase for data sync, cloud storage, and authentication.
- ✓ TensorFlow Lite or similar for future AI model deployment.

D. UI/UX Requirements

- ✓ Minimalist UI with clear navigation.
- ✓ Dashboard with live metrics.
- ✓ Easily accessible "Give Feedback" button.
- ✓ Speed test accessible in one tap.
- ✓ Visual rewards tracker (gamified experience).

- ✓ Onboarding screens explaining permissions.

5. Traceability Matrix

User Feedback Insight	Derived Requirement
"App shouldn't disturb me"	Smart prompt logic for feedback
"Want bonuses"	Rewards system for actions
"Who has the best network here?"	Location-based provider recommendations
"App must be light and easy"	Lightweight design with efficient background use
"Should test speed easily"	One-tap network speed testing

Identify Inconsistencies, Ambiguities, Missing Info

Area	Issue	Suggested Clarification
Feedback Prompting	"Prompt only when metrics degrade"	Define precise thresholds per metric
AI Integration	"AI recommends provider"	Clarify if based on static rules or a trained model
Reward Issuance	"Airtime bonuses"	Confirm delivery method: via operator APIs or manually
Metrics Frequency	Not specified	Suggest default values: e.g., every 15 min with 1 hr cap
Background Mode	Needs battery optimization	Define constraints to avoid excessive drain
Data Retention	Unclear policy	Set expected retention duration and anonymization rules

Prioritization of Requirements

Requirement	Priority	Phase
Real-time metric logging	High	MVP
Manual feedback with auto-data	High	MVP
Speed test	High	MVP
Smart feedback prompts	High	MVP
Provider recommendation	Medium	MVP
Reward system (basic point tracker)	High	MVP
AI-based recommendation	Low	Post-MVP
Reward redemption integration	Medium	Post-MVP
Privacy controls	High	MVP
Historical data logs	Medium	MVP

6. Summary of Requirement Analysis Outcomes

Step	Outcome
Elicitation	UserData.csv analyzed, key insights extracted
Clarification	Ambiguities converted to clear, testable behaviors
Prioritization	MVP defined with core functionality
Classification	Functional, Non-functional, Technical, and UI/UX categories established
Feasibility	All core features found technically implementable
Traceability	All requirements traceable to user or stakeholder input

3. REQUIREMENTS

After carefully analysing the data gathered from our Stakeholders and other mobile network users, we were able to come out with the following functional and non-functional requirements that will meet user needs or user requirements.

3.1 Functional Requirements:

Functional requirements define what the system should do. These are derived directly from user expectations, project goals, and operational needs. Each requirement includes a description of the intended behavior and the technologies proposed to fulfill it.

Requirement	Constraint	Description
Background Operation	Must not drain battery or interrupt user	The app runs quietly in the background using WorkManager (Android) and Background Fetch (iOS).
User Feedback Collection	Feedback triggered only if metrics are low	Prompts only when poor performance is detected. Also allows voluntary feedback anytime. Integrates with notification system to ensure non-intrusiveness.
Network Metrics Logging	Requires access to location and network status	Automatically collects: network type, signal strength, latency, jitter, packet loss, bandwidth, GPS location, timestamp, and service provider.
Feedback Submission	Data must be timestamped and tied to metrics	Users describe issues manually; app auto-fills location, timestamp, and network stats. Logs are stored in PostgreSQL/MongoDB .
Speed Test Tool	Requires real-time bandwidth measurement	One-click tool to test upload/download speed and ping using open-source or third-party libraries (e.g., Speedtest SDK).
Incentive System	Rewards must be monitored and fraud-proof	Tracks user contributions and assigns points or airtime. Requires secure user ID mapping. Integrated with reward redemption logic.
Network Provider Recommendation	Must reflect current location and real-time stats	Based on crowdsourced data; uses ML models to suggest better operators nearby.
Data Upload & Storage	Requires strong internet connection or caching	Syncs data to cloud using REST APIs with failover for offline mode. Secure cloud services like Firebase or AWS S3 are recommended.
Anomaly Detection (AI)	Needs sufficient device-level processing or cloud	Detects sudden network drops using models like Isolation Forest or Autoencoders . Used to trigger feedback.

Experience Prediction (AI)	Must ensure fast predictions	Predicts QoE from current metrics. Uses lightweight models like XGBoost or TensorFlow Lite for on-device inference.
Issue Categorization (AI)	Requires NLP model and labeled data	Categorizes textual feedback (e.g., "no signal", "slow data") using DistilBERT/BERT , fine-tuned for mobile-friendly performance.
Reward Optimization (AI)	Must avoid annoying the user	Predicts optimal moments to prompt users for feedback using context-aware data (location, time, app usage).

3.2 Non – Functional Requirements:

Non-functional requirements describe how the system performs its functions. They are quality attributes such as performance, usability, reliability, and technical design constraints. This section also includes platform dependencies and interface design principles. These non-functional requirements will also comprise of the technical and UI/UX requirements of the system.

Requirement	Constraint	Description
Performance	App size < 10MB, battery usage < 2% daily	Optimized background tasks, data batching, and minimal polling. Use of WorkManager (Android) and Background Fetch (iOS) to conserve system resources.
Security & Privacy	Full compliance with GDPR-like policies	All user data is encrypted in transit and at rest. Data is anonymized unless explicitly shared. Explicit user consent is required for all sensitive permissions.
Capacity & Scalability	Support 10,000+ concurrent users	Backend built on Node.js/FastAPI , auto-scaled cloud infrastructure (Firebase/AWS), and horizontally scalable databases like PostgreSQL/PostGIS or MongoDB .
Correctness	Data logging must be verifiable and precise	Timestamps, GPS tagging, and multi-point validation of logs. Accuracy in metrics computation (bandwidth, latency) ensured via reliable SDKs/APIs.
Efficiency	Minimize battery, bandwidth, and CPU usage	Data collection frequency is adaptive. All uploads are compressed and batched. Use of efficient data serialization (e.g., Protobuf or Gzip-compressed JSON).

Flexibility	Should support future feature additions easily	Modular and loosely coupled architecture. API-first design enables easy integration of new AI models, plugins, and data types.
Integrity	Logs must be free from corruption	All logs verified via hash/checksum mechanisms. Built-in retry logic and data validation routines.
Portability	Cross-platform mobile support	Built using Flutter to support both iOS and Android. Backend services are containerized with Docker and deployable on any cloud provider.
Maintainability	Easy to test, debug, and update	Modular code structure with separation of concerns (e.g., API layer, service layer). Full documentation and CI/CD pipelines for automatic testing and deployment.
Reliability	Uptime > 99.9%	Cloud infrastructure with high availability zones and fallback services. Logs stored offline and synced later to ensure no data loss.
Availability	Must function even without internet	Local storage (e.g., Hive/SQLite) caches logs and feedback for later synchronization when internet becomes available.
Testability	High coverage with easy mocking	Unit tests, integration tests, and test stubs for network conditions and AI feedback. Logs support full debug traceability.
Compliance	Must meet local telecom and data privacy regulations	Aligns with Cameroon's regulatory requirements and international privacy practices (e.g., GDPR). Includes user right to delete data.

Technical Requirements:

Technical Requirement	Constraint	Description
Platform	Must support Android and iOS	Flutter is the main framework. Optionally integrate Kotlin/Java for Android-specific background processing optimizations.
Permissions Needed	Must comply with OS-level privacy policies	Location (for metrics and recommendations), network state (for logging), background execution (for passive data collection), internet (for sync and model queries).
AI Model Inference	On-device + Cloud fallback	Use TensorFlow Lite/ONNX for lightweight models on device; fallback to cloud inference for more complex models.

Backend API	RESTful architecture	FastAPI/Node.js handles requests from the app. Supports asynchronous operations and real-time updates.
Database	Scalable, supports geospatial queries	PostgreSQL + PostGIS for structured and spatial data; MongoDB for flexibility in feedback schema.
Cloud Infrastructure	Secure, scalable, and compliant	Firebase, AWS, or GCP for hosting, authentication, and data storage. Built-in security and scalability capabilities.

UI/UX Requirements

UI/UX Requirement	Constraint	Description
Minimalistic Dashboard	Must show only necessary real-time metrics	Clearly displays key metrics: signal strength, latency, network type, and bandwidth. Uses intuitive icons and color indicators for quick comprehension.
Feedback Form	One input + auto-filled fields	Displays current location, timestamp, provider, and network state automatically. One free-text field for issue description. One-click submit.
Gamified Rewards Page	Must not be addictive or misleading	Shows progress bar, total points, and redemption options (e.g., airtime). Friendly animations for motivation but avoids gambling-style gamification.
Speed Test UI	Clear indication of results	One-tap interface with animated indicators. Shows upload, download, and ping with icons and colors. Optional history view for past results.
Recommendation Page	Must be localized and interactive	Map-based or list-based ranking of network providers around current location. Shows ratings, latency, and user reviews.

Settings Page	Must support preferences for privacy & notifications	Let's users toggle data sharing, feedback frequency, background data collection, and push notification preferences.
Onboarding Screens	Max 3–5 pages	Concise intro slides that explain app purpose, request permissions, and showcase key features before first use.
Accessibility	Must support screen readers and large fonts	Full support for OS accessibility APIs. Font scaling and visual contrast adapted to user preferences.

The outlined functional and non-functional requirements form a robust foundation for developing a reliable, efficient, and user-centric mobile application for mobile network experience monitoring in Cameroon. By combining advanced technical specifications—such as cross-platform development, background processing, real-time metric logging, and AI-based analytics—with carefully considered UI/UX and compliance strategies, the system is designed to not only meet the current needs of end users but also adapt to future growth, technological evolution, and regulatory shifts.

The inclusion of detailed technical and design constraints ensures that the system remains lightweight, secure, and scalable, while the clear user interface and gamified engagement model drive usability and retention. These requirements have been derived from comprehensive user feedback, practical feasibility analysis, and industry best practices, ensuring both functional completeness and design integrity.

Together, these specifications will guide implementation, validation, and eventual deployment, enabling the delivery of a high-impact mobile tool that empowers users and improves overall telecom service quality through participatory data collection and intelligent recommendations.

This system is not just a feedback app — it evolves into a smart, user-centric QoE platform powered by AI to optimize user engagement and data quality while providing tangible benefits to users. Key differentiators include:

- Passive feedback triggers via anomaly detection.
- Real-time location-aware network provider recommendations.
- Lightweight and low-battery impact design.
- AI-powered predictions to enhance service quality understanding.

4. PROJECT CONSTRAINTS:

4.1 Time Constraints:

We face a significant constraint with a tight timeline of two months. To address this, we'll prioritize features, adopt agile methodologies, and implement sprints for incremental development.

4.2 Project Scope:

Given the time constraint, we need to focus on delivering core functionalities initially, with a plan for continuous improvement in subsequent phases.

4.3 User Adoption Constraints

The success of the application heavily relies on user participation and engagement, which may be influenced by factors such as user trust and the perceived value of providing feedback.

4.4 Technical and Integration Constraints

The application must integrate with existing backend systems and databases for data storage and analysis, which could introduce compatibility challenges. Reliance on third-party services (e.g., cloud storage, analytics tools) may introduce dependencies that affect project timelines and performance.

4.5 Budgetary Constraints

The project must stay within a specified budget, impacting the choice of technologies, tools, and resources used for development. Costs related to server maintenance, data storage, and user incentives must be considered to ensure the project's sustainability.

4.6 Regulatory Constraints

Compliance with data protection laws and telecommunication regulations is necessary, particularly concerning user consent for data collection and the handling of personally identifiable information (PII).

5. ESTIMATED COST

Category	Description	Estimated Cost
----------	-------------	----------------

Survey Heart Google Form	Used to collect feedback and for requirement gathering	Free
Google and AI Chat Bots	Used for research	
StarUML	Used for Requirement Analysis (UML diagrams and ER diagrams)	
Figma	Used for Interface design	
Microsoft Word	Used for Interface design and documentation	
PowerPoint	Used for Presentation	
VS Code	Used as a text editor for coding	
Local host server	Used for testing	
React JS and Firebase	For designing and implementing database	
Data Bundle	For internet access	25,000 FCFA +
SaaS and SDK Fees	Third Party APIs	50,000 FCFA +
Server Hosting Costs	Used to host the App	200,000 FCFA annually

6. PROJECT CALENDAR

TASK	DESCRIPTION	COMMENCEMENT DATE	COMPLETION TIME
Requirement Gathering	Information was gathered using: ❖ Questionnaires ❖ Interview Session with Stakeholders	08/ 04/ 2025	15/ 04/ 2025 STATUS: DONE TOTAL TIME:

	❖ Reverse Engineering		7 Days.
Analysis of Requirement	-Review and analysis of requirements gathered -Identification of inconsistencies, ambiguities, and missing information -Prioritizing requirements based on importance and feasibility -Classification of requirements (functional and non-functional) -Development of the software requirement specification (SRS)	31/ 04/ 2025	05/ 05/ 2025 STATUS: ONGOING TOTAL TIME: 6 Days

MVP (Minimum Viable Product)

Objective: Launch a lightweight, core-functionality app to collect user feedback and network metrics passively and actively.

Feature	Description	Why in MVP?
Background Service	Automatically collects network metrics periodically.	Core to your promise of passive QoE data collection.
Auto-detect Location	Capture user location for context.	Essential for geo-tagging data and building regional insights.
Network Metrics Logging	Collect signal strength, network type, latency, jitter, packet loss.	Baseline dataset for analysis and future ML use.
Manual Feedback	Let users describe their	The heart of active user

Submission	experience (issue description).	input.
Timestamp & Service Provider Detection	Auto-record current time and detect active carrier.	Completes the data context around feedback.
Simple Admin Dashboard (Web)	View submitted data, download CSV reports.	Needed for early-stage analytics and debugging.
Lightweight Authentication (Optional)	Simple login with phone number or anonymous mode.	Prevents abuse, enables unique user tracking.
Lightweight App Size (Under 15MB)	Focus on minimal design and libraries.	Keeps adoption high in low-end Android markets.

Deployment Tools: Flutter (for lightweight, cross-platform app), Firebase or Supabase (for quick auth + backend), OrientDB for the final database.

Phase 2 – Core Enhancements

Objective: Increase user engagement and data quality.

Feature	Description
Prompted Feedback (Smart)	Ask for feedback only when metrics fall below threshold.
Speed Test Tool	Let users run speed tests and log results.
Rewards System (Prototype)	Give small points or badges for submitted feedback.
Passive Notifications	Notify user of network changes or low quality.
Basic Coverage Map	Show which provider is best in user's location (static at first).
Settings Page	Let users choose how often to be prompted, mute prompts, etc.

Phase 3 – Full Version with AI & Personalization

Objective: Use data to deliver intelligent, location-aware, and incentive-driven features.

Feature	Description
AI Model: Predict Dissatisfaction	Train model on past metrics + feedback to predict likelihood of poor QoE without needing manual input.

Real-time Recommendation	Operator	Suggest best provider in current location using AI+data.
Smart Prompt Engine		AI determines when to prompt, based on patterns from user history.
Airtime/Data System	Rewards	Connect with APIs of providers to reward users with data/airtime.
Dynamic (Crowdsourced)	Heatmap	Show live performance heatmap from all user data.
Network Quality Trends		Show historical performance of user's current provider.

MVP Clickable Prototype Wireframe Overview

◆ 1. Home Screen

Elements:

- App logo & name.
- Tagline (e.g., "Know Your Network. Improve Your Experience.").
- Quick summary card: current provider, signal strength, and quality badge.
- Nav bar or icons for: Dashboard | Feedback | Speed Test | Rewards | Settings.

◆ 2. Smart Network Metrics Dashboard

Elements:

- Real-time display of:
 - ✓ Network type (4G, 5G, etc.)
 - ✓ Signal strength (bars or dBm)
 - ✓ Latency, jitter, packet loss, bandwidth
- AI Health Indicator: Summary status (e.g., "Good / Poor / Unstable").
- Smart Prompt Status: "You may be asked for feedback if network degrades."
- Refresh metrics button.

◆ 3. Feedback System (Smart & Manual Modes)

a) Smart Feedback Prompt (Triggered Automatically)

- Appears only when network metrics drop below thresholds.
- Short survey: “How satisfied are you right now?” (1–5 stars).
- Optional: “What went wrong?” (dropdown + optional text input).
- "Skip" or "Later" options available.

b) Manual Feedback Form

- Accessible via button on home/dashboard.
- Elements:
 - ✓ Text input: issue description.
 - ✓ Auto-filled read-only fields:
 - Timestamp.
 - Location.
 - Network provider.
 - Current network metrics.
 - ✓ Submit button with confirmation.

◆ 4. Speed Test Tool

Elements:

- “Test My Speed” button.
- Live animated indicators for:
 - ✓ Download speed.
 - ✓ Upload speed.
 - ✓ Ping.
- Historical log with timestamp and provider.

◆ 5. Provider Recommendation System

Elements:

- Auto-location detection.
- “Recommended Provider in Your Area” card.
- Button: “See More Providers” → list ranked by QoE in area.
- Map view (static for MVP): shows colored zones per provider.

◆ 6. Rewards & Incentives

Elements:

- Points dashboard:
 - ✓ Points from passive data sharing.
 - ✓ Points from feedback submissions.
- Redemption center (simple): “Earn X points to get 100MB bonus.”
- Progress bar toward next reward.
- “How to earn points” info card.

◆ 7. Settings & Preferences

Elements:

- Notification preferences (on/off prompts, frequency).
- Data collection frequency (every 15 mins / 1 hour / custom).
- Manual sync toggle (e.g., Wi-Fi only).
- Privacy notice + data usage terms.
- Contact support.

◆ 8. Background Passive Logger (Visual Component)

Elements:

- Silent operation.
- Status shown via a persistent notification: “Monitoring Network Quality...”
- Toggle in settings to enable/disable background collection.

◆ 9. Bonus: User Onboarding Flow (Optional but Valuable)

Elements:

- 3-slide intro on:
 1. What the app does.
 2. How your data helps.
 3. How you earn rewards.
- Permissions request (location, background access, network stats).

AI Model Placeholder (For Future Phase)

- In MVP: show a “Predicted Network Health” badge (random or static demo).
- Backend: Begin collecting data to train model (feedback + metrics pairing)