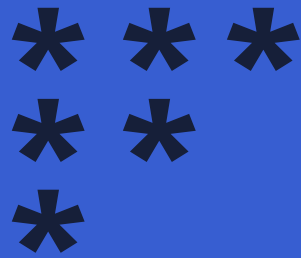# System Modelling and Design of a Mobile network experience monitoring App for Network subscribers

Design and Implementation of a Mobile App for Collection of Users' Experience Data

Course: CEF 440

Task 4 Presentation

Group Members: [Names]

Instructor: [Instructor Name]

"Presented by Group 23"

* Purpose :Provides Visual Understanding

* Why It Matters: Helps design, implement, and communicate system logic

* Diagrams Included:
  - Context Diagram
  - Data Flow Diagrams (Level 1 & 2)
  - Use Case Diagram
  - Sequence Diagram
  - Class Diagram
  - Deployment Diagram

Purpose

Why it matters

Diagrams like Use Case , Context e.t.c

Includes Diagrams
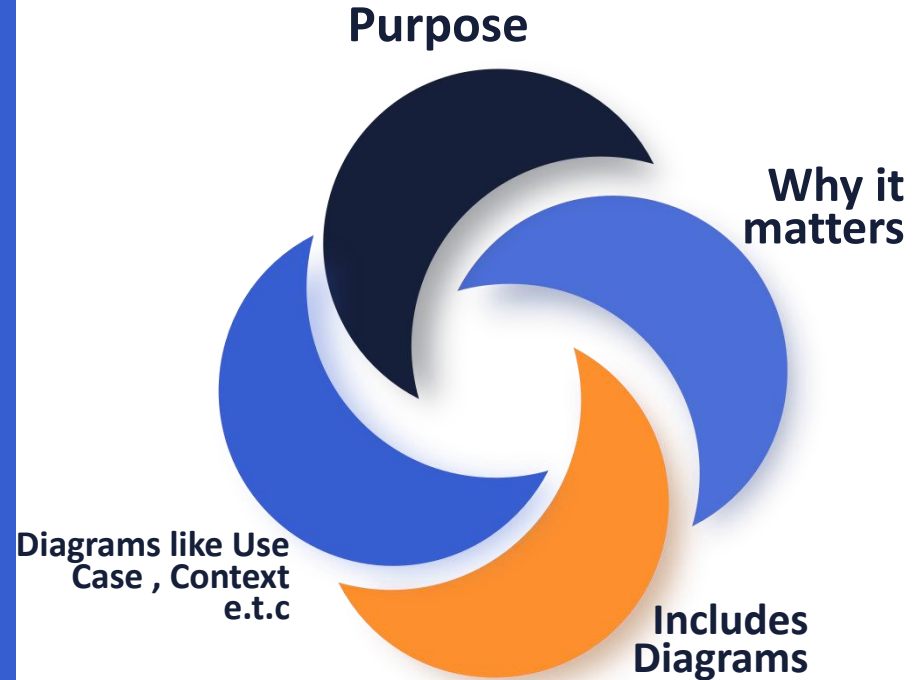
• Displays interaction between the mobile app and external entities

• Entities: User, Admin/Analyst, Backend Server
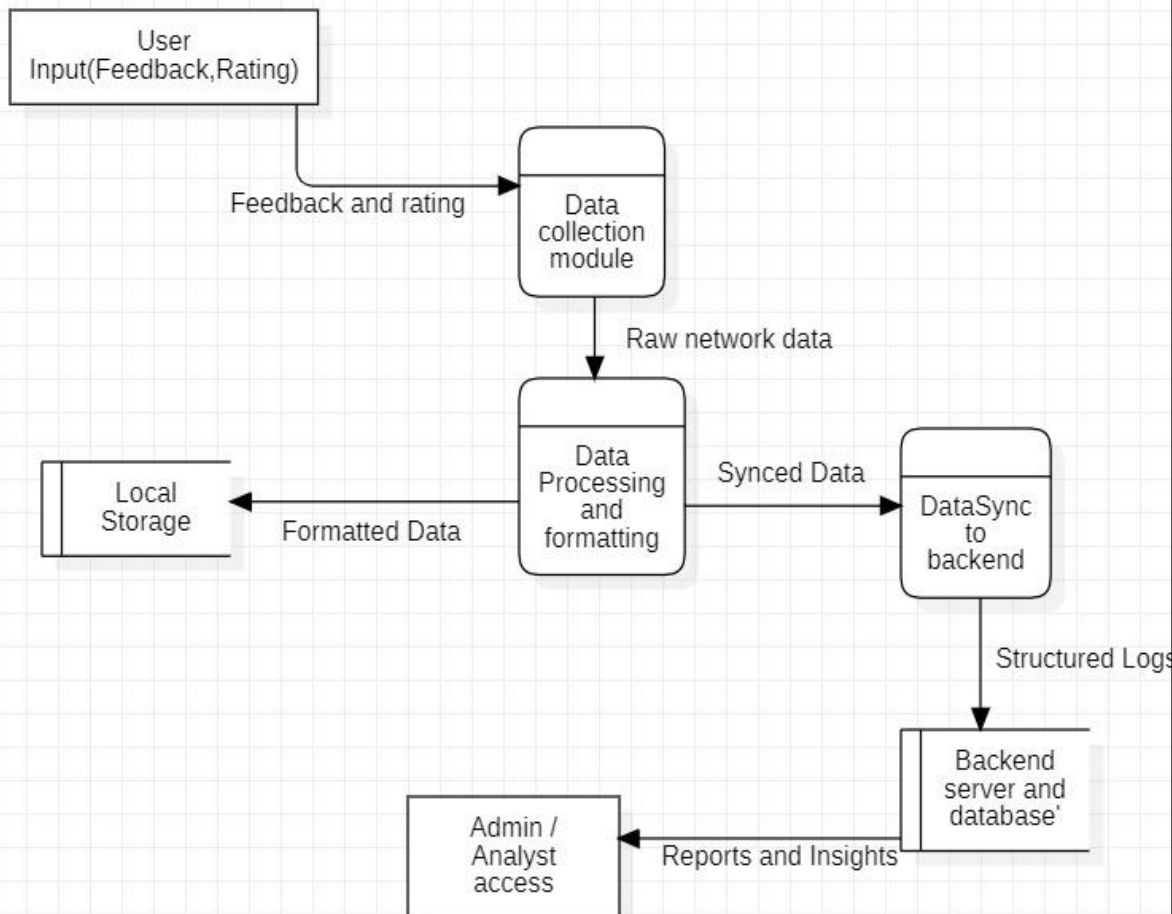
• Core system: Mobile App for QoE Data Collection
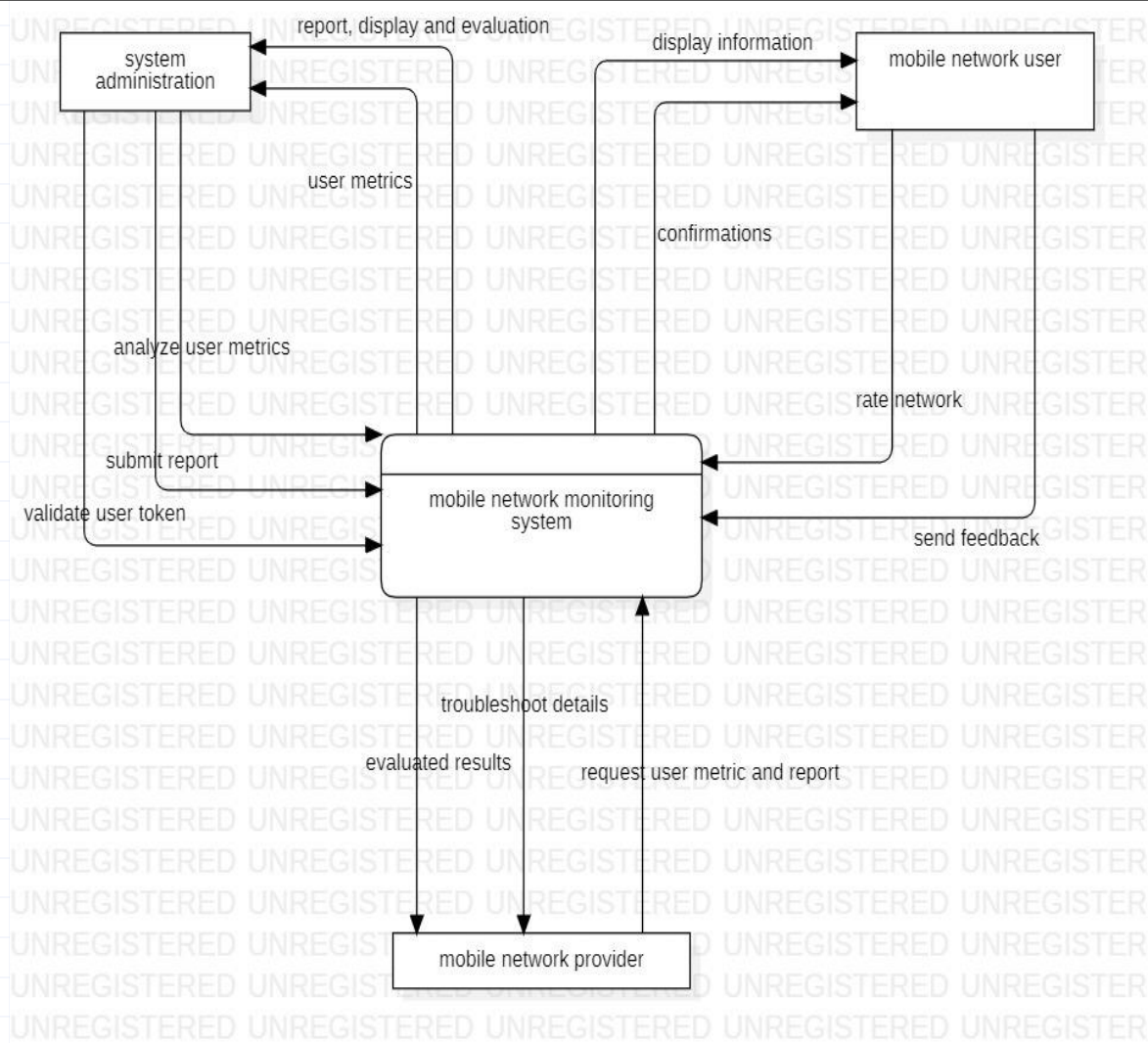
Shows high-level process flow

• Key processes:
  - Data Collection
  - Data Processing
  - Sync to Backend

• Data stores: Local Storage, Backend DB
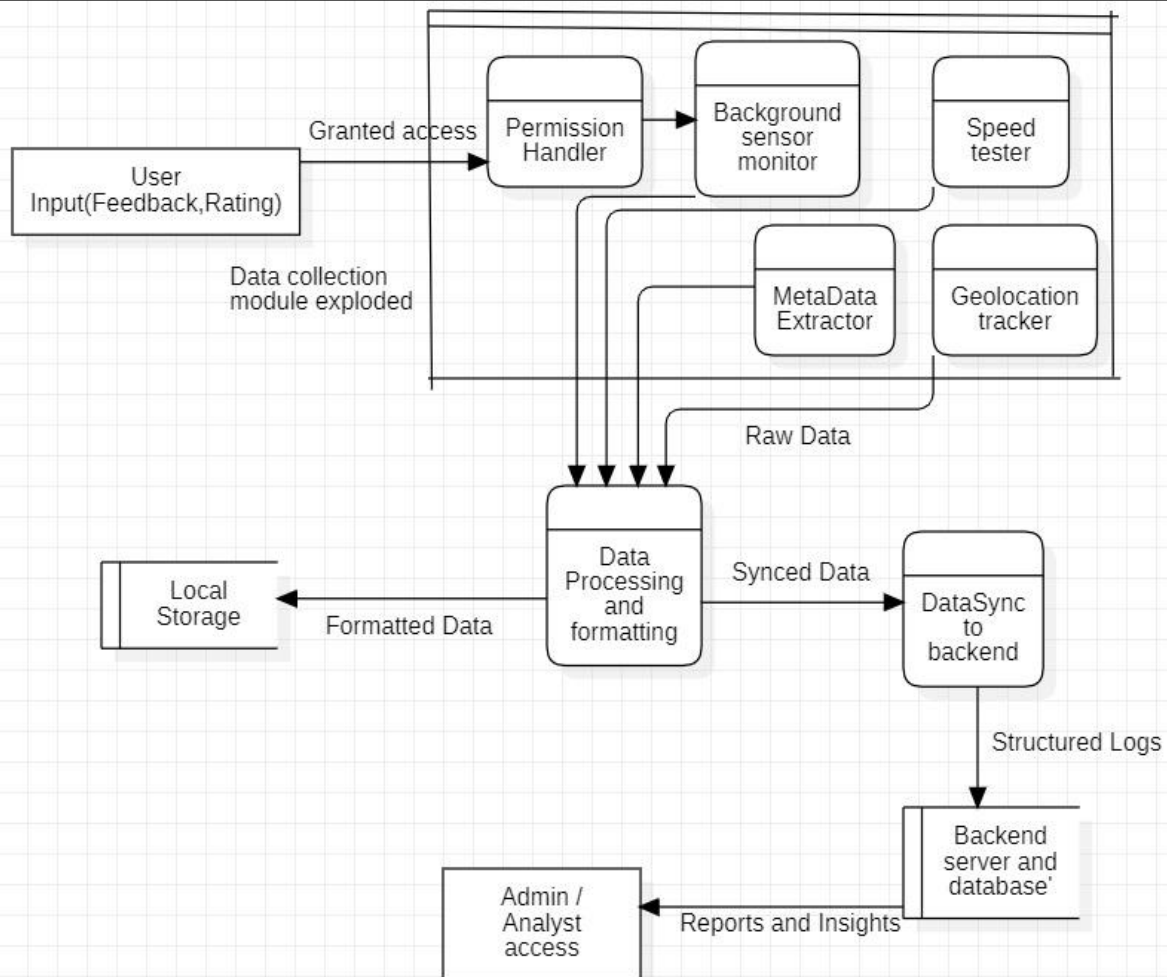
DFD Diagram level 2

-Explodes 'Data Collection Module'

Includes:
  - Permission Handler
  - Background Sensor Monitor
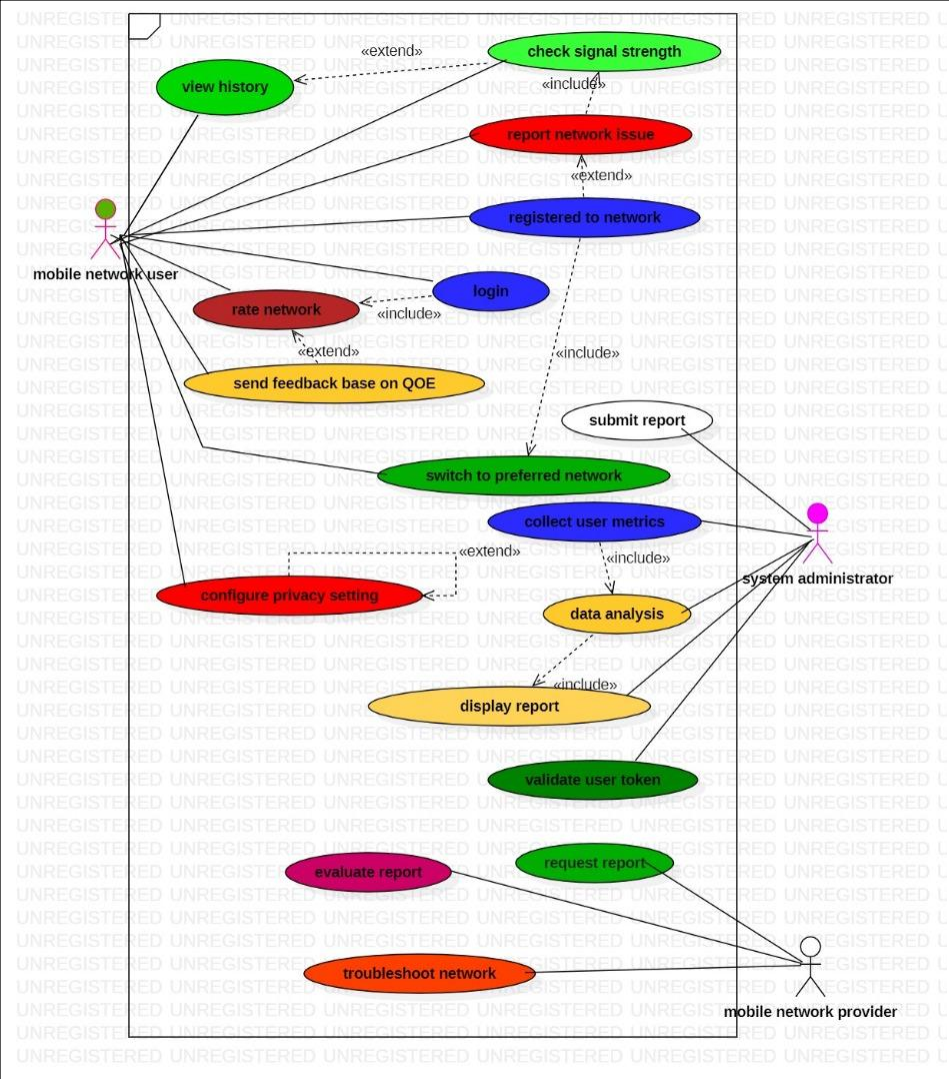 - Speed Tester
  - Metadata & Geo-location
Tracker

USE CASE DIAGRAM

* Actors: User, Admin, Mobile network provider

* Use Cases:
  - Submit Feedback
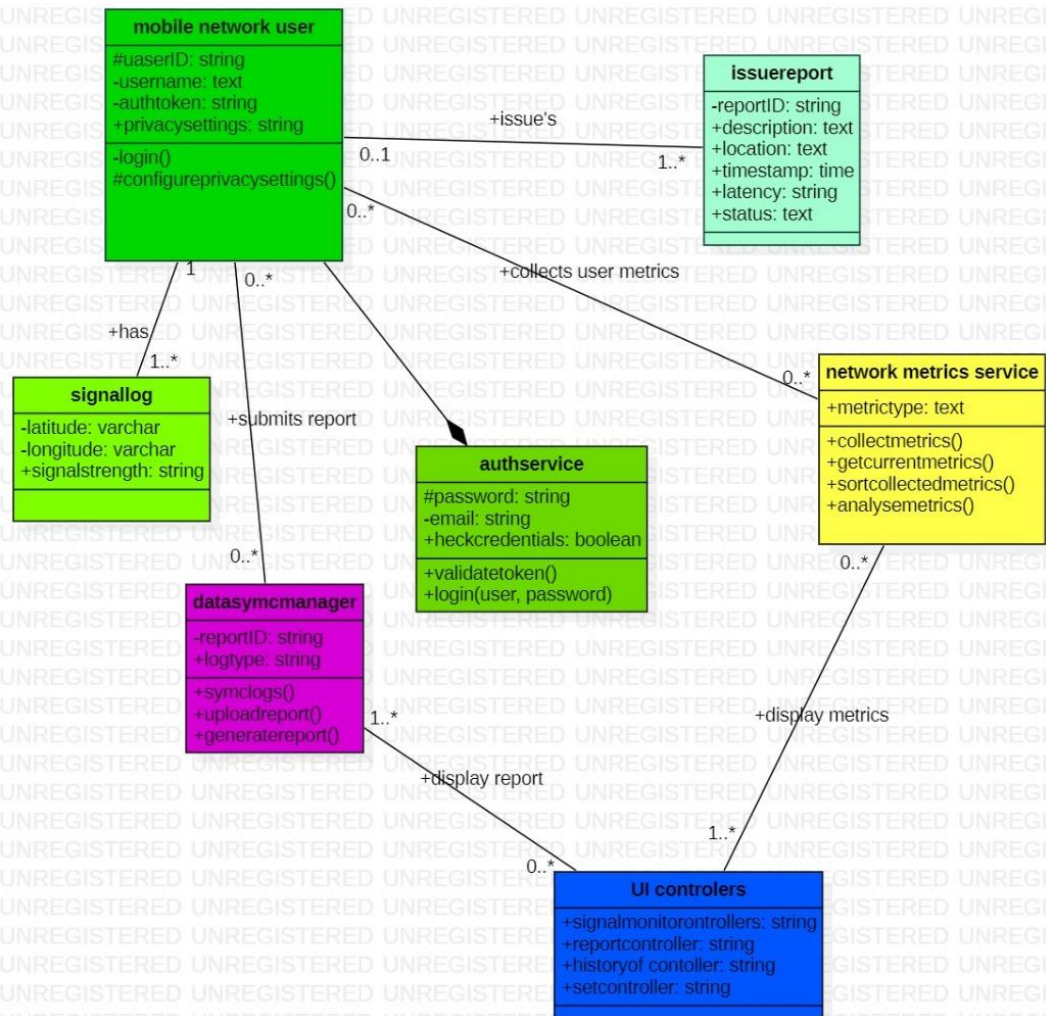 - View History
  - Analyze Reports

CLASS DIAGRAM

* Main Classes:
  - User
  - Feedback
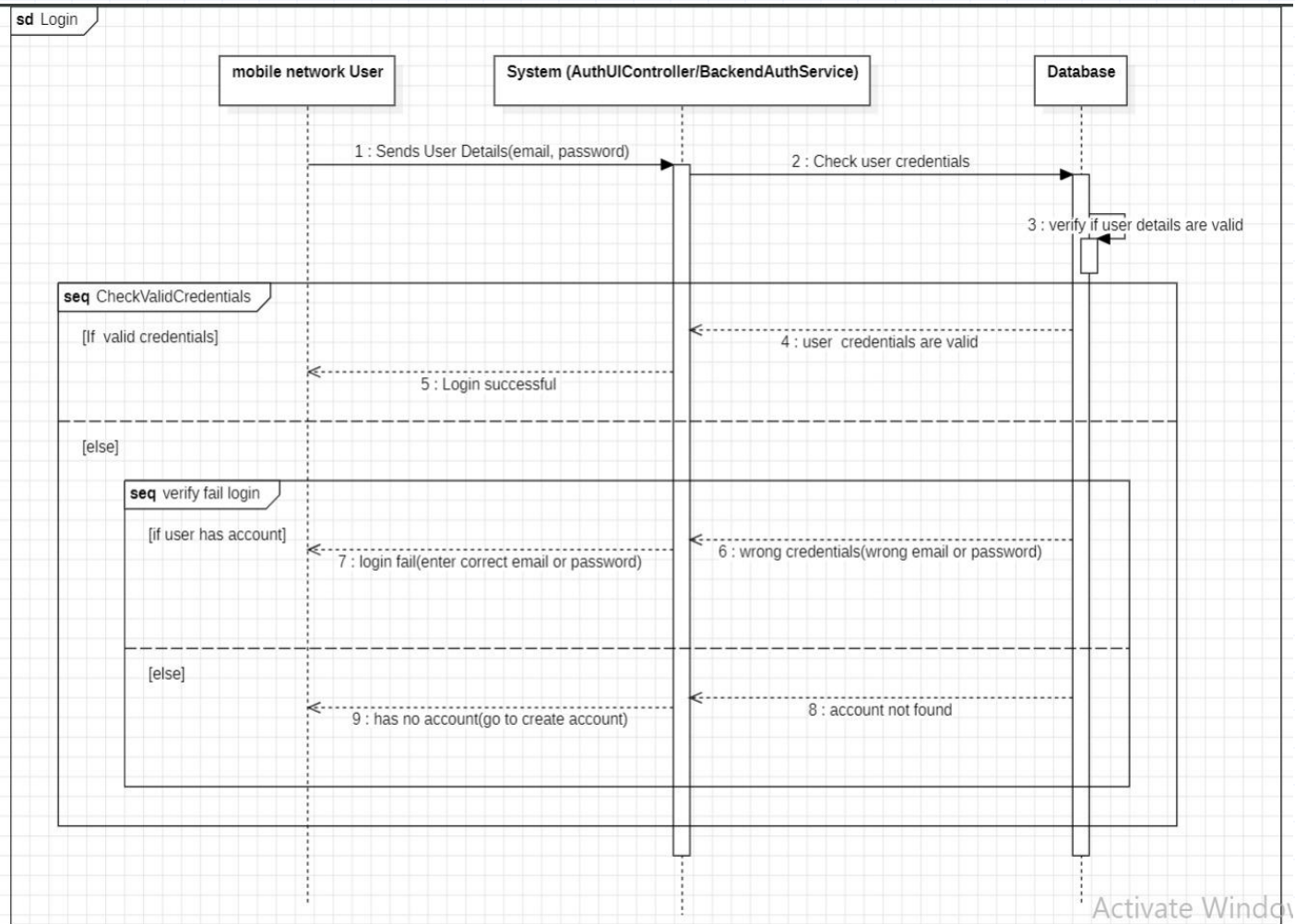 - NetworkMetrics
  - Report

* Relationships: Associations, Aggregations

mobile network user
#uaserID: string
-username: text
-authtoken: string
+privacysettings: string

-login()
#configureprivacysettings()

issuereport
-reportID: string
+description: text
+location: text
+timestamp: time
+latency: string
+status: text

+issue's
0..1   1..*

0..*

+collects user metrics

1   0..*

+has
1..*

signallog
-latitude: varchar
-longitude: varchar
+signalstrength: string

+submits report

network metrics service
+metrictype: text

+collectmetrics()
+getcurrentmetrics()
+sortcollectedmetrics()
+analysemetrics()

0..*

authservice
#password: string
-email: string
+heckcredentials: boolean

+validatetoken()
+login(user, password)

0..*

datasymcmanager
-reportID: string
+logtype: string

+symclogs()
+uploadreport()
+generatereport()

1..*

+display metrics

+display report

0..*

1..*

UI controlers
+signalmonitorontrollers: string
+reportcontroller: string
+historyof contoller: string
+setcontroller: string

Sequence Diagram

- Demonstrates flow of operations

- Example: Submit Feedback → Format → Validate → Sync

sd SequenceDiagram1

mobile network user

DataSyncManger

Database

2 : submit issue

1 : check if authenticated

seq Check if authenticated

[if login]

3 : calls submitIssueReport(data)

4 : Stores the issueReport

6 : issue successfully saved

5 : report saved

[else]

7 : redirects to create account or login page

Shows deployment Architecture

We also took into account components such as mobile device, Backend Server and Admin Terminal

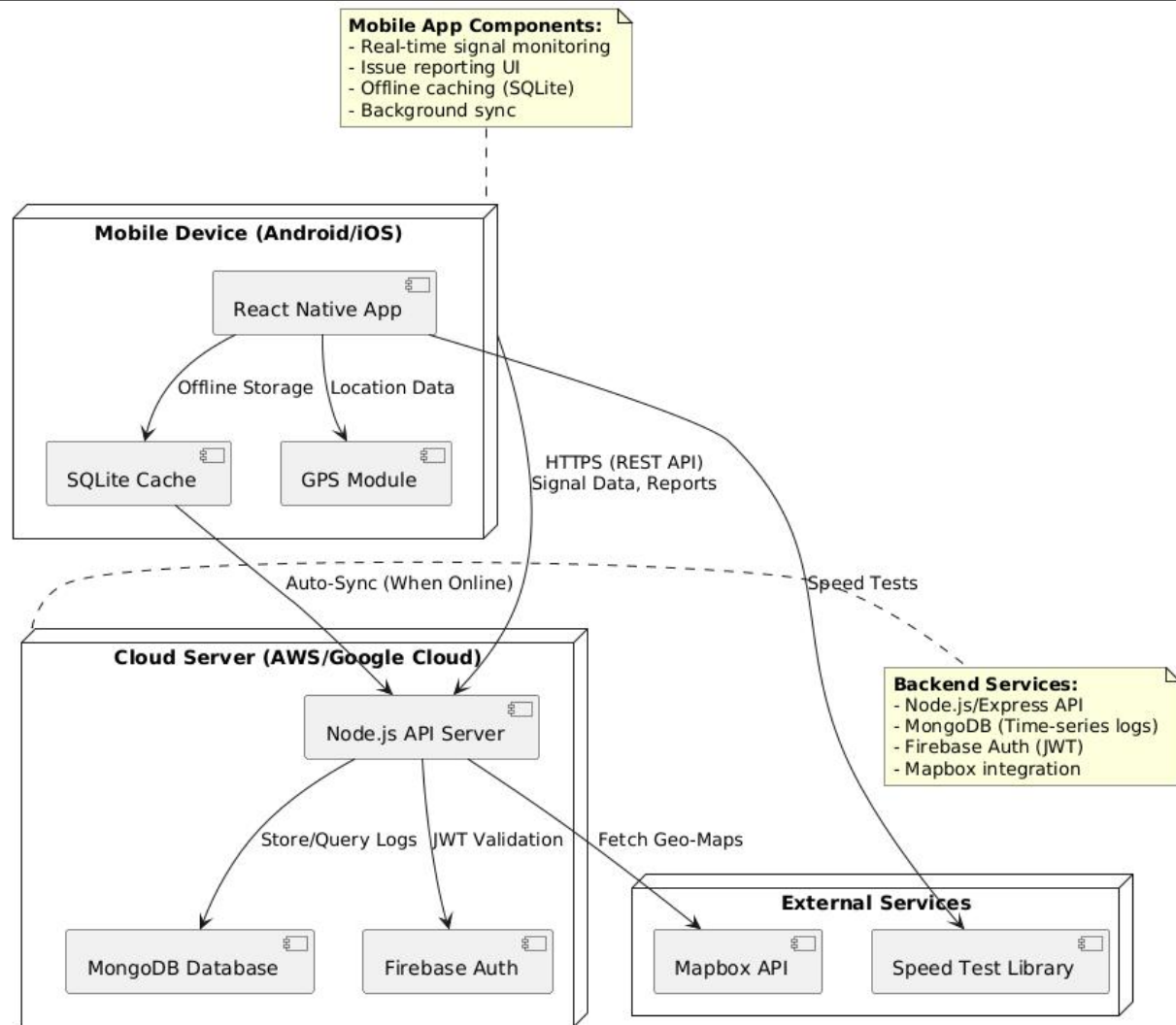**Key Technologies were addressed and identified e.g React Native, Node.js, Firebase**

Deployment Diagram

Mobile App Components:
- Real-time signal monitoring
- Issue reporting UI
- Offline caching (SQLite)
- Background sync

Mobile Device (Android/iOS)

React Native App

Offline Storage — Location Data

SQLite Cache

GPS Module

HTTPS (REST API)
Signal Data, Reports

Auto-Sync (When Online)

Speed Tests

Cloud Server (AWS/Google Cloud)

Node.js API Server

Backend Services:
- Node.js/Express API
- MongoDB (Time-series logs)
- Firebase Auth (JWT)
- Mapbox integration

Store/Query Logs   JWT Validation   Fetch Geo-Maps

MongoDB Database

Firebase Auth

External Services

Mapbox API

Speed Test Library

01

02

03

Models improve system design understanding

Provide clarity for developers and stakeholders

Next Steps:
  - Full implementation
  - User Testing and Deployment

Thank you