Noise - SNR Collector Progress Report

Version <1.0>

2015719243

Cem ŞANAL

02/11/2017

CONTENT

1. PROJECT OVERVIEW

1.1. Project Name: Noise - SNR Collector

1.2. Introduction

Noise – SNR Collector is a user oriented and mathematical ANDROID application. Mobile network signal levels will be understood via finding signal-to-noise ratio. Users could see signal levels via their mobile phones by walking on their destination and they could see signal levels in different areas from the data fetched from servers. SNR data will be saved according to its location. Users could see other users SNR data. Signal analysis and network speed assumptions could be performed. Routes could be drawn.

1.3. Project Definition

Briefly, Noise meter for ANDROID Mobile Phones. In this project, the aim is to design and development of signal to noise ratio (SNR) measuring software for ANDROID based mobile phones. The volume of sounds using the ANDROID system will help to measure the noise (Noise Meter). An apparatus for comparing sound intensity levels usually in decibels. Mandatory features: Average, min, max and peak signal to noise ratios in dB will be shown with a nice interface. Users would draw routes from the map. Software will determine and show the best and the worst path according to signal levels. Every created route path will be determined according to users SNR data.

2. FUNCTIONAL REQUIREMENTS

1.1. Calculation Requirements

1.1.1. The signal-to-noise ratio (SNR) of the connected mobile network should be calculated. – FREQ01

1.1.2. The peak signal-to-noise ratio (PSNR) of the connected mobile network should be calculated. – FREQ02

1.1.3. The average signal-to-noise ratio of the connected mobile network should be calculated. – FREQ03

1.1.4. The minimum signal-to-noise ratio of the connected mobile network should be calculated. – FREQ04

1.1.5. The maximum signal-to-noise ratio of the connected mobile network should be calculated. – FREQ05

1.1.6. New route from current or clicked location to the destination should be calculated. – FREQ06

1.2. User Interface Requirements

1.1.7. The interface should be user friendly. – FREQ07

1.1.8. The user would see a graph which compares sound intensity levels. – FREQ08

1.1.9. Sound intensity levels would be shown in decibels. – FREQ09

1.1.10. Sound intensity levels would also be shown in the other required formats. – FREQ10

1.1.11. The signal-to-noise ratio (SNR) should be shown in the graph in the main activity. – FREQ11

1.1.12. The peak signal-to-noise ratio (PSNR) should be shown in the graph in the main activity. – FREQ12

1.1.13. The average signal-to-noise ratio should be shown in the graph in the main activity. – FREQ13

1.1.14. The minimum signal-to-noise ratio should be shown in the graph in the main activity. – FREQ14

1.1.15. The maximum signal-to-noise ratio should be shown in the graph in the main activity. – FREQ15

1.1.16. The SNR data would be shown on a map with colors while Noise – SNR Collector is working in a separated activity. – FREQ16

1.1.17. The adjusted route should be shown on the same map while Noise – SNR Collector is working in a separated activity. – FREQ17

1.1.18. The graph should be refreshed in at least 2 seconds interval. – FREQ18

1.3. Data Processing Requirements

1.1.19. Individually saving signal to noise ratio data based on location data and time. – FREQ19

1.1.20. Sound intensity levels should be saved as decibels. – FREQ20

1.1.21. The Signal-to-noise ratio should be calculated and saved on the decibels scale. – FREQ21

1.1.22. The average signal-to-noise ratio should be calculated and saved on the decibels scale. – FREQ22

1.1.23. The minimum signal-to-noise ratio should be calculated and saved on the decibels scale. – FREQ23

1.1.24. The maximum signal-to-noise ratio should be calculated and saved on the decibels scale. – FREQ24

1.1.25. Saved SNR and location data should be sent to the application server regularly. – FREQ25

1.1.26. MYSQL database via Amazon RDS would be used as database server. – FREQ26

1.1.27. Saved SNR data should be fetched regularly by each client. – FREQ27

1.4. Users' Profile Requirements

1.1.28. A user account should have a mail address. – FREQ28

1.1.29. A user account would have an id. – FREQ29

1.1.30. A user account would have a password. – FREQ30

3. NON-FUNCTIONAL REQUIREMENTS

1.1. System

1.1.1. The project should be operated in the ANDROID operating system. – NREQ01

1.1.2. The ANDROID system should be between 4.0 to 8.0 versions. – NREQ02

1.1.3. When a new software update arrives, user should not use the old version. – NREQ03

1.1.4. Software should work as a background service whether or not program will be closed. – NREQ04

1.2. Performance

1.2.1. **(Response Time <= 1sec):** The system should be able to respond to a request of a user in not more than a second. – NREQ05

1.2.2. **Throughput:** Multi-tasking should be enabled to allow multiple users, initially at least 10 simultaneous users should interact with the system without having to wait others to finish their work with the system. – NREQ06

1.3. Scalability

1.3.1. Simultaneous users should be increased gradually after the first release. – NREQ07

1.4. Capacity

1.4.1. Initially at least 10 simultaneous users should interact with the system. – NREQ08

1.5. Maintenance

1.5.1. When a potential problem or a security threat has been recognized inside the system, maintenance might be performed as an update. – NREQ09

1.6. Availability

1.6.1. The system should be available day and night except the maintenance times and when updates are needed. – NREQ10

1.7. Security

1.7.1. The system should be secure for the server side. – NREQ11

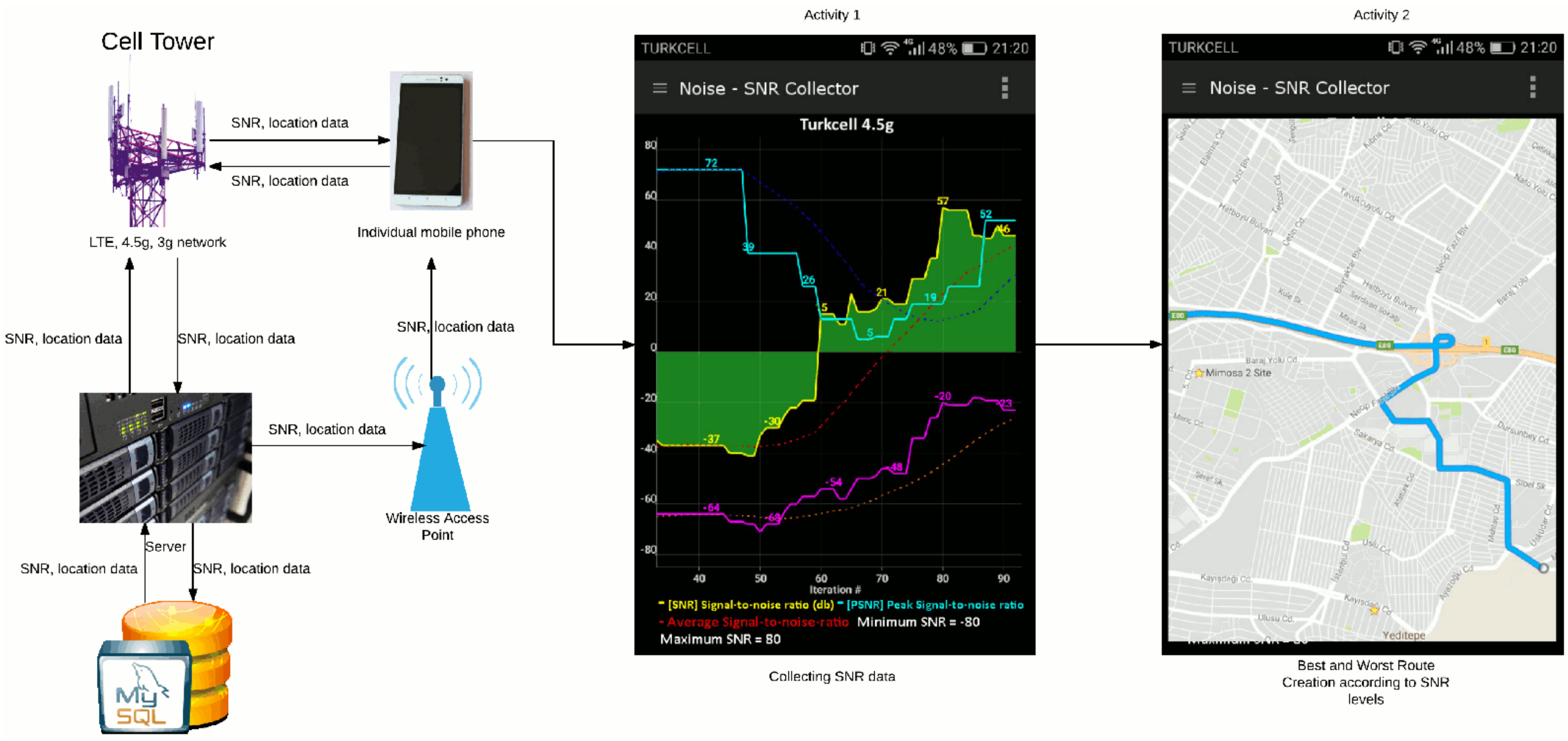
1.7.2. The system should be secure both for the client side. – NREQ12

1.8. Standards

1.8.1. Decibels should be used as the standard format. – NREQ13

1.8.2. Other formats should be changed to decibels before saving. – NREQ14

4. DESIGN


Cell Tower
SNR, location data
SNR, location data
LTE, 4.5g, 3g network
Individual mobile phone
SNR, location data
SNR, location data
SNR, location data
SNR, location data
Wireless Access Point
Server
SNR, location data
SNR, location data
Activity 1
TURKCELL 48% 21:20
Noise - SNR Collector
Turkcell 4.5g
80
60
40
20
0
-20
-40
-60
-80
40
50
60
70
80
90
Iteration #
[SNR] Signal-to-noise ratio (db)
[PSNR] Peak Signal-to-noise ratio
Average Signal-to-noise-ratio
Minimum SNR = -80
Maximum SNR = 80
Collecting SNR data
Activity 2
TURKCELL 48% 21:20
Noise - SNR Collector
Mimoza 2 Site
Yeditepe
Best and Worst Route Creation according to SNR levels

1.1. System Components

1.1.1. Users Mobile phone

1.1.2. Cell Tower (LTE, 4.5g, 3g, 2g)

1.1.3. Server

1.1.4. MYSQL Database

5. GLOSSARY

1.2. SNR

Signal-to-noise ratio (abbreviated SNR or S/N) is a measure that compares the level of a desired signal to the level of background noise. SNR ratio which is the ratio of signal power to the noise power, generally shown in decibels. SNR is measured as in the formula below:

$$\mathrm{SNR}_{\mathrm{dB}} = \left(A_{\mathrm{signal,dB}} - A_{\mathrm{noise,dB}}\right)$$

1.1. PSNR

The psnr function implements the following equation to calculate the Peak Signal-to-Noise Ratio (PSNR):

$$PSNR = 10\log_{10}(peakval^{2}/MSE)$$

where peakval is either specified by the user or taken from the range of the image datatype (e.g. for uint8 image it is 255). MSE is the mean square error, i.e. MSE between A and ref.

1.2. Average SNR

Average SNR will be understood from real tests.

1.3. Minimum SNR

Minimum SNR will be understood from real tests.

1.4. Maximum SNR

Maximum SNR will be understood from tests.