SOFTWARE REQUIREMENTS SPECIFICATION

for

General Network Access Tester

Version 1.0

Prepared by Michael Vessia Jr.

May 10, 2016

Contents

1 Introduction

1.1 Purpose

General Network Access Tester is an Android application which helps IT departments, network administrators, or anyone that wants to monitor a network that they have access to. The application should be free to download on the Google Play store, and the application will be open source. The application provides information that will make it easier for someone to troubleshoot their network. The application also reduces the workload on IT departments by allowing them to troubleshoot more quickly.

This document is meant to give an overview of the features of the General Network Access Tester, making the process of using the application for one's own network simple, and making contributing to the project easy and painless.

1.2 Document Conventions

The General Network Access Tester will hereafter be referred to as GNAT. Users will be someone that interacts with the mobile application. An administrator will be the person who owns the network and is likely having the logs sent to them. The administrator can and likely will also be the user.

1.3 Intended Audience and Reading Suggestions

This document is meant for administrators who would like to have their network monitored by GNAT, or developers who would like to make contributions to GNAT. It is not required that users read this document, but if they would like further clarification about the settings they they are configuring within the application, they will find that information in this document. Developers will be interested in the source code. As of version 1.0, the source code can be found at https://github.com/MichaelVessia/gnat.

1.4 Project Scope

GNAT is targetting network administrators who have mobile devices connecting to their network and would like more information about connection attempts, connection drops, and other network issues. It should reduce the time an IT team needs to troubleshoot why people are unable to connect to their network.

2 Overall Description

2.1 Product Perspective

GNAT should integrate well with third party software as the logs are in JSON format, but ultimately should be a stand-alone application and not depend on any other software. More functionality is unlocked if the administrator has a server that accepts JSON POST requests.

GNAT's output log files should integrate well into an existing monitoring/dashboard system, but this is not required for the application to function.

2.2 Product Functions

Within the mobile application, users will be able to enter a number of configuration options that will affect what kind of logs are generated. The connection info will be logged in the background at a user specified interval. Although the application tests network access, it ultimately assumes that the user has access to the internet. If there is no wifi network to connect to, functionality will be extremely limited.

The application needs internet (WIFI) to generate the desired logs and send them to the server.

If the administrator uses a dashboard service that has an API, it should accept JSON POST requests, and they can enter the url into GNAT's settings. If it does not, they are responsible for creating some kind of web service that can act as a middle man.

2.3 User Classes and Characteristics

The user of the application is expected to either be the administrator, or be a knowledgeable user that recieved the relevant information from the administator. End users of the network should not have to interact with GNAT, but assuming they had the relevant information nothing would be stopping them.

2.4 Operating Environment

GNAT will run on the Android operating system. The application will require the user to have at least Android 4.0.3 (Ice Cream Sandwich). It also assumes there is a network to be tested.

Outside of the application, the administrator should feel free to create an internet facing web page, and place a file on that page. The app will attempt to connect to this web page, based on the address the user enters in the configuration. GNAT can simply check if the file was able to be downloaded, or the file can be a script that does something else and stores more detailed logs on the server.

2.5 Design and Implementation Constraints

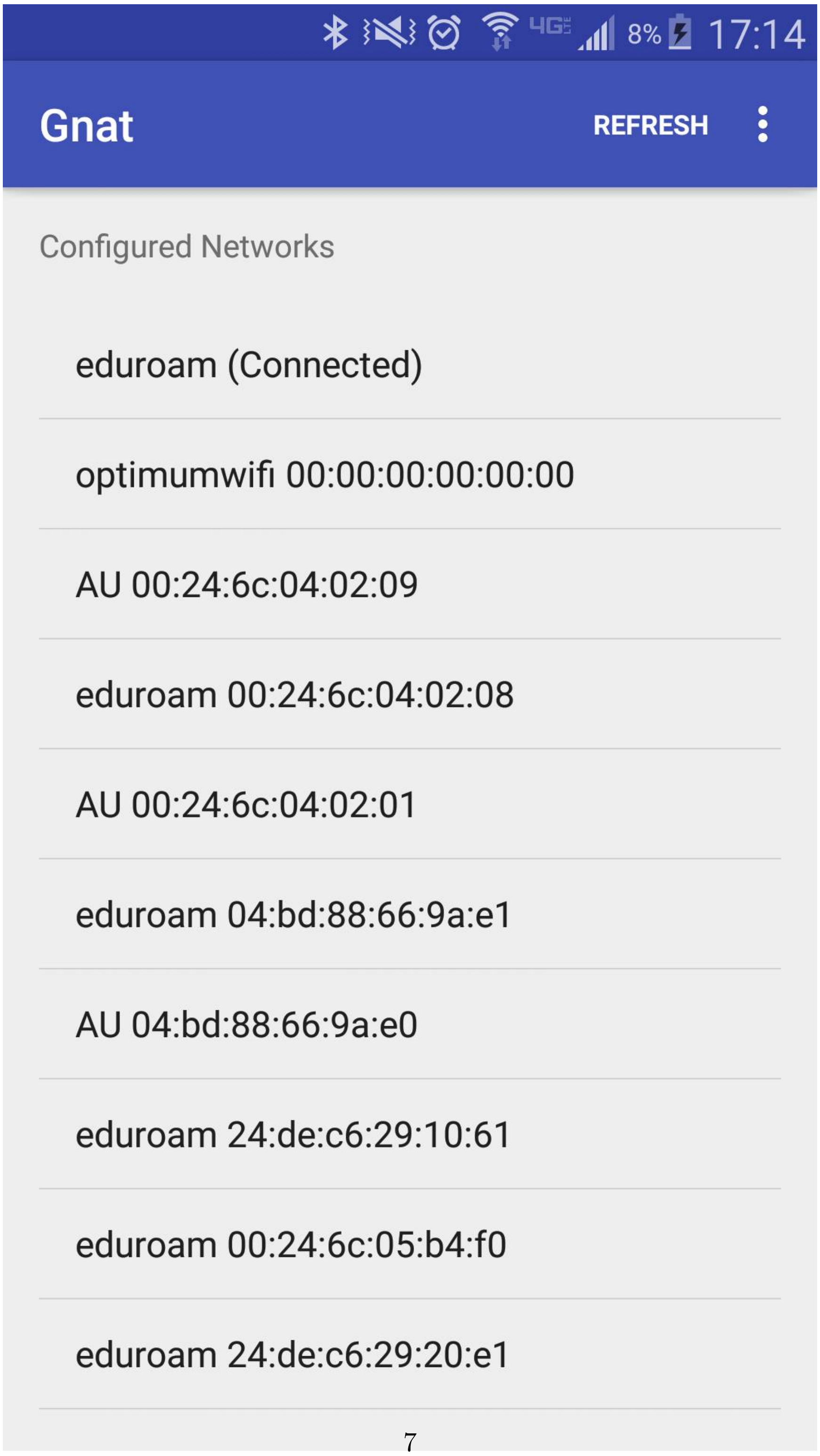
- GNAT will write JSON logs to local device storage, there should be room on the device. It will not delete any old logs, this is up to the user.
- GNAT will send POST requests to the server specified. This server needs to be properly configured and must exist.

2.6 Assumptions and Dependencies

- If a user plans to use a third party monitoring system, they need to make sure they are credentialed to do so and that the service provides an API. If the service has not yet been used with GNAT, a user should feel free to add their anonymized script to the repository, and create a pull request at https://github.com/MichaelVessia/gnat.
- GNAT assumes the user knows the password to the network they are trying to test. Failure to connect should be from other issues, not authentication.

3 External Interface Requirements

3.1 User Interfaces



Configured Networks

eduroam (Connected)

optimumwifi 00:00:00:00:00:00

AU 00:24:6c:04:02:09

eduroam 00:24:6c:04:02:08

AU 00:24:6c:04:02:01

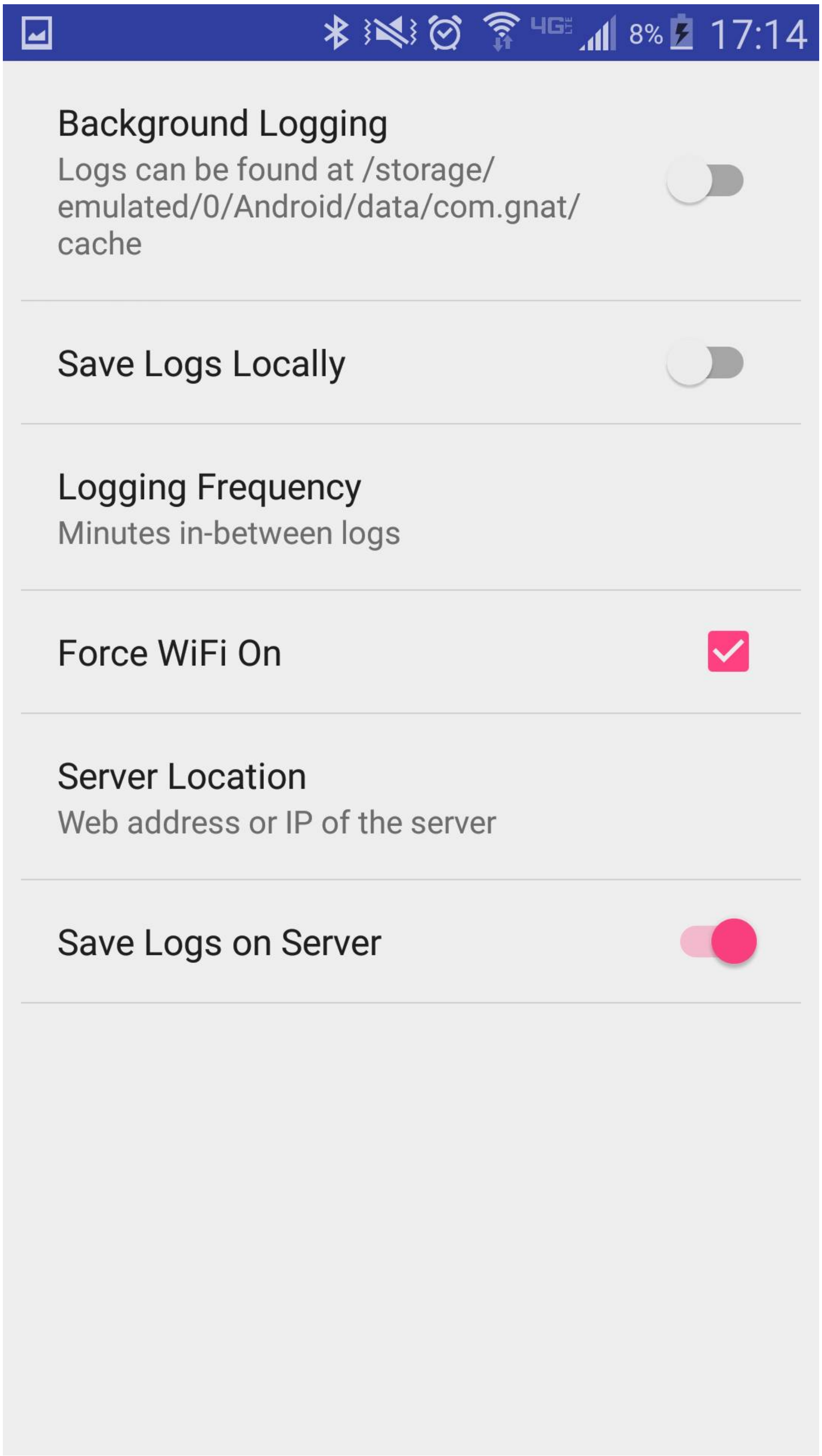
eduroam 04:bd:88:66:9a:e1

AU 04:bd:88:66:9a:e0

eduroam 24:de:c6:29:10:61

eduroam 00:24:6c:05:b4:f0

eduroam 24:de:c6:29:20:e1



Background Logging
Logs can be found at /storage/emulated/0/Android/data/com.gnat/cache ○

Save Logs Locally ○

Logging Frequency
Minutes in-between logs

Force WiFi On ☑

Server Location
Web address or IP of the server

Save Logs on Server ●

3.2 Software Interfaces

A sample web app for companion use with GNAT exists at https://github.com/MichaelVessia/gnat-rails. It was built with Ruby on Rails and PostgreSQL. Administrators should feel free to create their own using their own server/technology stack.

3.3 Communications Interfaces

Logs are sent via HTTP POST request using the application/json header.

4 System Features

4.1 Generation of Log Files

4.1.1 Description and Priority

Priority: High

Necessary for the desired purpose of the application. Without the log files, GNAT's use is limited.

4.1.2 Stimulus/Response Sequences

1. User connects to the network
2. User enters configuration, enables logging
3. Log file is generated at specified interval

4.1.3 Functional Requirements

- Log Files should be in JSON format to make the integration with APIs easier in the future.

4.2 Connection to Administrator's Server

4.2.1 Description and Priority

Priority: High

If there is no web page to connect to, then we can only monitor connection failure and nothing beyond that. Having a server with a web page allows us to offload log storage onto the server as well.

4.2.2 Stimulus/Response Sequences

1. User enters appropriate configuration options and begins logging.
2. JSON object is generated containing log information.
3. HTTP POST request is made to server.

4.2.3 Functional Requirements

- Log files are stored on the server
- Log files should be sent through an API to the IT monitoring software if it is provided
- If a network connection is unable to be established, log files should be stored locally

4.3 Software Quality Attributes

One of GNAT's key goals is to be useful to anyone, regardless of their network configuration or external software that they use. For this reason, it is necessary that it remain open source so that if someone wants to make it useable for their system, they can implement the necessary changes to do so.

4.4 Appendix A: Glossary

GNAT: General Network Access Tester
HTTP: Hypertext Transfer Protocol
HTTPS: Hypertext Transfer Protocol Secure
API: Application Program Interface
JSON: JavaScript Object Notation
XML: eXtensible Markup Language
app: Application
OS: Operating System