# SOFTWARE REQUIREMENTS SPECIFICATION

# for

# General Network Access Tester

Version 1.0

Prepared by Michael Vessia Jr.

February 17, 2016

# Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| 21 | 22 | 23 | 24 |
| 31 | 32 | 33 | 34 |

# 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 a considerable amount of mobile devices connecting to their network, and would like more information about failed connections. 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 into an existing monitoring/dashboard system, but ultimately should be a stand-alone application and not depend on any other software.

## 2.2 Product Functions

Within the mobile application, users will be able to enter a number of configuration options. The network connection that will be attempted will be based upon these configuration options. The result of this connection attempt will be logged if anything goes wrong. Although the application tests network access, it ultimately assumes that the user has access to the internet. If there is an initial connection error, logs will be stored locally until a connection is made, at which point it will send the logs via the specified means of communication.

The application needs both internet and GPS connection to generate the desired logs and send them to the specified location. The GPS will allow the logs to contain relevant geographical information, like where the device was when it attempted to connect from.

If the administrator uses a dashboard service that has an API, GNAT will be able to send the logs via the API to be viewed in the dashboard service.

## 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.

## 2.5 Design and Implementation Constraints

- GNAT will be accepting a network password as user input. This should be handled appropriately.
- GNAT will be able to connect to a third-party dashboard monitoring service like Zenoss, it should handle these credentials correctly as well.
- GNAT will connect to the network using the HttpUrlConnection client, provided by the Android platform.

## 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 functionality to GNAT, 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

No UI as of 1.0.

## 3.2 Software Interfaces

GNAT will be able to make use of third party IT monitoring systems. The communication between GNAT and the external software will likely be dependent on the particular software. As of 1.0there are no such integrated components.

## 3.3 Communications Interfaces

GNAT will connect to the network using the HttpUrlConnection client, provided by the Android platform. It will send messages through API's likely using JSON or XML, and can send log files to servers using HTTP.

# 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

- User enters configuration
- User connects to the network
- Some error occurs when attempting to visit web page or download resource from web page
- Log file is Generated

### 4.1.3 Functional Requirements

- Log Files should be in a JSON or XML if at all possible, to make the integration with API's easier in the future.

## 4.2 Sending Log Files to Administrator

### 4.2.1 Description and Priority

Priority: High

If log files can only be stored locally and not sent to the administrator, then GNAT's use is limited.

### 4.2.2 Stimulus/Response Sequences

- User enters configuration
- User connects to the network
- Some error occurs when attempting to visit web page or download resource from web page

- Log file is Generated
- Log file is sent to administrator based on configuration

### 4.2.3 Functional Requirements

- Log files should be sent through an API to the IT monitoring software if it is provided
- Log files should also be sent directly to the admin's server via HTTP for backups if they configure GNAT to do so
- If a network connection is unable to be established, log files should be stored locally

# 5 Other Nonfunctional Requirements

## 5.1 Security Requirements

Security concerns are mostly related to the proper storage and communication of network passwords and API keys. Measures will be taken to make sure that these are secure. Connections to send log files will be made over HTTPS when possible as well.

## 5.2 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.

## 5.3 Appendix A: Glossary

- GNAT - General Network Access Tester
- HTTP - Hypertext Transfer Protocol
- API - Application Program Interface
- JSON - JavaScript Object Notation
- XML - eXtensible Markup Language
- app - Application