

**CSCI321 – Project (Android Packet Sniffer)**

Group Member List

|  |  |
| --- | --- |
| Name | Student ID |
| Soh Yu Xuan | 5498636 |
| Timothy Chin | 5498399 |
| Kenneth Huang | 5498442 |
| Kendrick Tan | 5026556 |

2. Introduction

Computers used to be large clunky machines which were the innovation of the IT sector but now, almost everyone if not all, has a “small computer” with them all the time, a smartphone. One of the duties of a network administrator is to monitor the network. This can be done with a computer with either with an external network card or with an internal one. However, the computer was something that was needed, adding weight and hassle. What if one could monitor and sniff packets from their device whom they carried with them everywhere? One could simply be in a location, capturing packets in their vicinity for analysis, without having to lug around a laptop right into their mobile device.

What is packet sniffing?

“A sniffer (packet sniffer) is a tool that intercepts data flowing in a network. If computers are connected to a local area network that is not filtered or switched, the traffic can be broadcast to all computers contained in the same segment. This doesn’t generally occur, since computers are generally told to ignore all the comings and goings of traffic from other computers. However, in the case of the sniffer, all traffic is captured when the sniffer software commands the Network Interface Card (NIC) to stop ignoring the traffic. The NIC is put into promiscuous mode, and it reads communications between computers within a particular segment. This allows the sniffer to seize everything that is flowing in the network, which can lead to the unauthorized access of sensitive data. A packet sniffer can take the form of either a hardware or software solution.

2.1 Document Overview

The document describes the design process which is used in creating the application, it also includes UML Diagrams to illustrate the design of the system that can be used to show the functionalities of the application. Other than functionalities there will also be test cases to show that the application have the appropriate quality for use.

2.2 Reference Documents

(if any will be added inside)

3. Application Description

This section will give an overview of the application design, development, and Implementation

3.1 Introduction

(introduction of application)

3.2 Operational Examples

(this section will describe our application in a real scenario base)

3.3 System Requirements

(Requirements of our System e.g Android Platform, Version x, etc)

3.4 Functional Requirements

Under this section, we will be describing the functional requirements of our application.

3.4.1 Start/Stop Packet Sniffing

When users open the application, they will be able to select when they wish to start and stop the capturing of packets. The application works by utilizing a binary that will be started in a process with SuperUser permission. The process runs in a thread and stops only when the stop button is pressed. The packets are captured using the binary and are redirected to a text file.

3.4.2 Display Information

The application works by reading in the text file that is at the same time being written to by the sniffing thread. As data enters the text file, another thread is used to read in the packet data to the application. Another thread at the same time will get the data read in and display to the gui portion of the application.

Users will be able to view the packets captured by utilizing 3 threads, writing to a text file, reading a text file and updating to the display all being done concurrently.

3.4.3 Save Packets to File

After the user is done capturing packets, they will be able to save the data to a file which they can then use another application to view the data at their own time or for system administrators they will be able to keep records of the type of data that has been going through the network.

This packet information is saved in a ‘.pcap’ format contains more information that can be displayed on a mobile device. It would be better analyzed on a computer.

3.4.4 Filter of the packet

In this section our application would be designed to filter the types of packet that the user wishes to see, according to certain devices/websites/IP addresses/packet type.

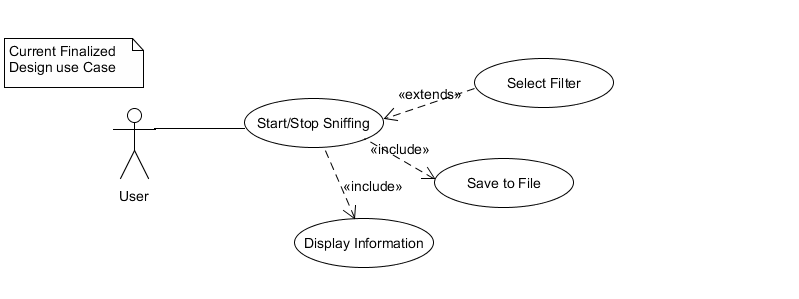
3.5 Non-functional Requirements (Quality of life Service)

This section describes the non-functional requirements of our application that is designed to help users navigate through the application with as little difficulty as possible

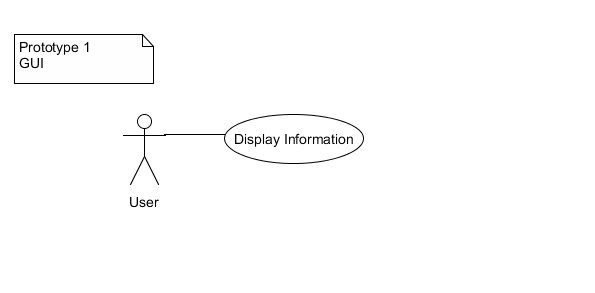
4. UML Diagram

4.1 Use Case Diagrams

4.1.1 First Iteration



First Design of Application



First GUI Design of Application

4.1.2 Second Iteration

(to be added)

4.2 Sequence Diagrams

(to be added)

4.3 Other Possible Diagrams

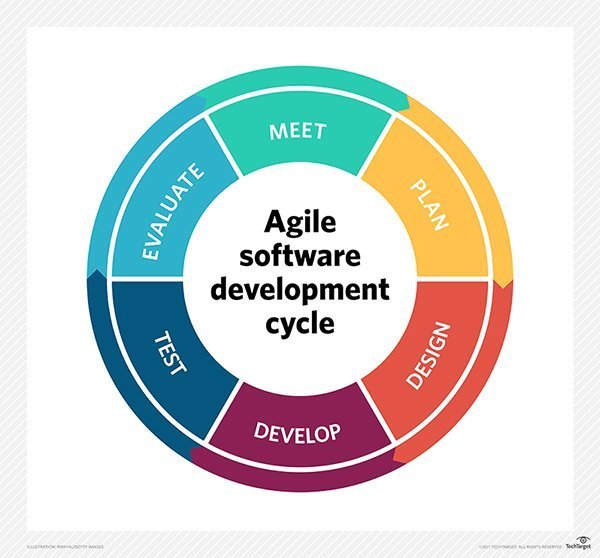
(to be added)

5. User Interface Designs

(application interface designs to be added)

6. Software Development

6.1 Software design process



Agile (Scrum) is used to develop our application, it provides the ability to create and respond to change to succeed in an uncertain and turbulent environment. Agile is the most applicable due to the unpredictability of the application due to rooting and specific modes based on the WLAN cards of the type of devices that we plan to incorporate into. With Agile we can makes changes as and when necessary to help overcome such issues.

6.2 Development Tools and Environment

* Android Studio
* UMLet
* Other possible tools we used

6.3 Schedule of project

6.4 roles & responsibilities

|  |  |
| --- | --- |
| Team Members | Roles |
| Soh Yu Xuan | Team Leader, Lead Programmer |
| Timothy Chin | Assistant Programmer / Lead UI |
| Kenneth Huang | Assistant Programmer / Lead Tester |
| Kendrick Tan | Project Management / Documenter |

6.5 Software Version Control

(to be added)

7. Application test

7.1 test cases

8. Conclusion