

Report

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

**Group SS18/2A Android Network Sniffer Report**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Number : <SS18/2A>** | | | |
|  | **Student Number** | **Name** | **Email Address** |
| 1 | 5498636 | Soh Yu Xuan(Team Leader) | yxsoh004@mymail.sim.edu.sg |
| 2 | 5498399 | Chin Tian Zhi Timothy | tztchin001@mymail.sim.edu.sg |
| 3 | 5498442 | Huang Shun Yang Kenneth | sykhuang001@mymail.sim.edu.sg |
| 4 | 5026556 | Tan Siyuan Kendrick | sktan024@mymail.sim.edu.sg |

Table of Contents

1. [1. Project Overview 1](#_Toc518727224)

[A) Vision](#_Toc518727225)

[B) Initial Use-Case Model](#_Toc518727226)

[C) Project Plan](#_Toc518727227)

1. [2. Roles and Responsibilities 2](#_Toc518727228)
2. [3. Risk Analysis and Counter Measures 3](#_Toc518727229)
3. [4. UML Diagrams 4](#_Toc518727230)

[Design Artefacts - Use Cases (Iteration 1)](#_Toc518727231)

[Design Artefacts – Activity Workflows (Iteration 1)](#_Toc518727232)

[Design Artefacts – Data Flow Diagram (Iteration 1)](#_Toc518727233)

[Design Artefacts - Class Diagrams (Iteration 1)](#_Toc518727234)

[Design Artefacts - Sequence Diagrams (Iteration 1)](#_Toc518727235)

1. [5. Test Plan Design 12](#_Toc518727236)

[<Relate to fuctional requirement>](#_Toc518727237)

[Club Membership Management - The system should allow the administrator to add new members](#_Toc518727238)

1. [6. Evidence of using VCS (Version Control Software) 14](#_Toc518727239)

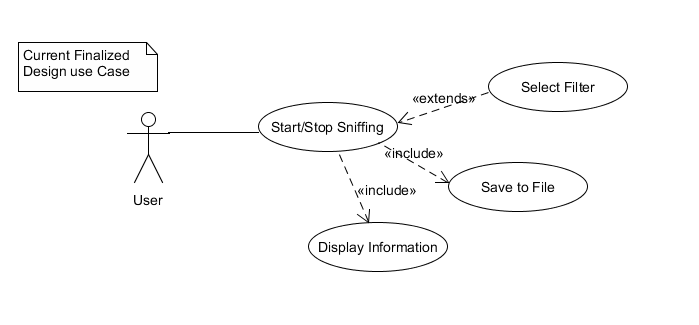
# 1. Project Overview

Designing of an Android Network Sniffer which in our team’s understanding is to achieve real-time scanning of network traffic and recording it to a file for later use. The application would be useful for people with Networking and Security background as the people who use this application would be System Administrators, Similar Application Developers and possibly general users. As such an application is already existing on computers our implementation of it onto a device like android has its limitation but it also thrills us to try and develop an actual working version of it.

## A) Vision

* Our key feature of the application would be real time capturing of packets and being able to see them displayed on the application.
* To be able to save packet data into a file for later view.
* After being able to achieve the first objective the second one would be to filter traffic based on certain criteria’s such as type of device or maybe specific type of traffic
* If possible be able to scan the access point for devices connected to it.

## B) Initial Use-Case Model



## C) Project Plan

* Due to the limitations of the Chipset our available device has been restricted and therefore plan to use Nexus 5 due to concerns such as device cost
* Due to the need to have root access, which may cause bricking to a phone we have opted to use older no longer in use devices
* The device needs to be in monitor/promiscuous mode our choice of device has been restricted.
* The rooting methods may or may not be suitable for our device so therefore we need to have alternate ways to get network traffic such as using VPN
* Using libraries from Pcap to assist with the project as well as tcp dump

# 2. Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| Team Number : < **SS18/2A** > | | | |
|  | **Student Name** | **Role** | **Artefacts** |
| 1 | Soh Yu Xuan | Team Leader, Lead Programmer |  |
| 2 | Chin Tian Zhi Timothy | Lead Programmer |  |
| 3 | Huang Shun Yang Kenneth | Android UI Programmer |  |
| 4 | Tan Siyuan Kendrick | Team Manager, Lead Documenter | Technical Document, Initial Use case, research on products, SRS, Report |

# 3. Risk Analysis and Counter Measures

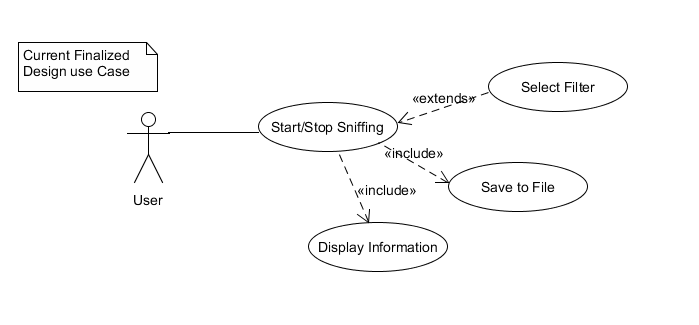
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Impact Type** | **Risk Seriousness (%)** | **Likelihood of Occurrence (%)** | **Risk Description** |
| 1 | Successful Task completion | 90% | 50% | Being able to complete rooting as it is the main criteria failing to do so will result in needing alternative methods which cause re-scope of project |
| 2 | Deadline | 90% | 10% | If deadline is not met all team members might fail the project |
| 3 | Unable to display on phone’s application | 50% | 20% | If application does not display correctly the bulk of the project might not work needing more time to debug |
| 4 | Unable to obtain device for project | 90% | 20% | If we are unable to have a device that meet our criteria |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Risk Description** | **Proposed Mgmt Plan** | **(Possible) Reduction in Risk Seriousness (%)** |
| 1 | Unable to root phone | Plan 1 : Search for alternative methods like VPN | -50% |
| Plan 2 : use alternate rooting method till successful | -25% |
| 2 | Unable to obtain device for project | Plan 1: Buy from people who are selling old phones | 75% |
| 3 |  |  |  |

# 4. UML Diagrams

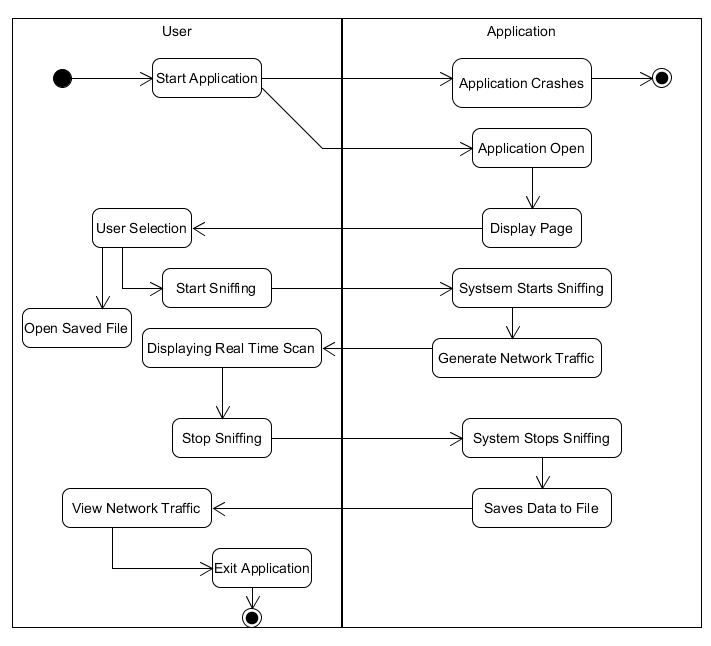
## Design Artefacts - Use Cases (Iteration 1)

Main use case diagram:

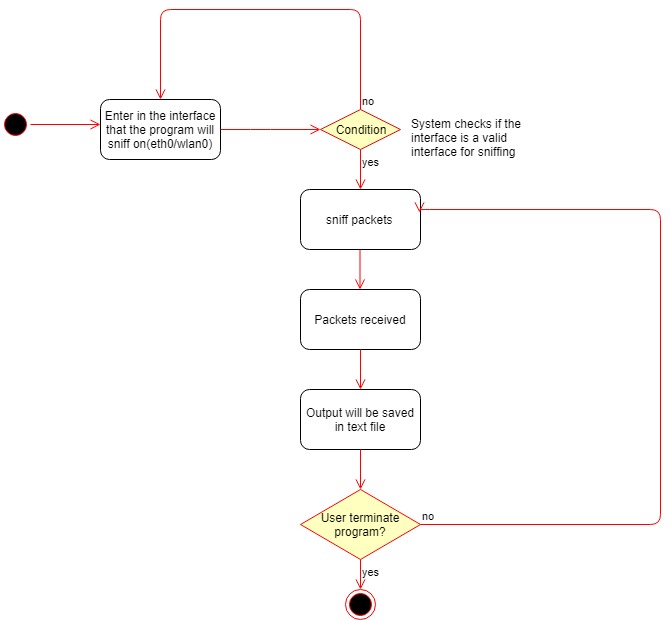


## Design Artefacts – Activity Workflows (Iteration 1)

Android app activity diagram:

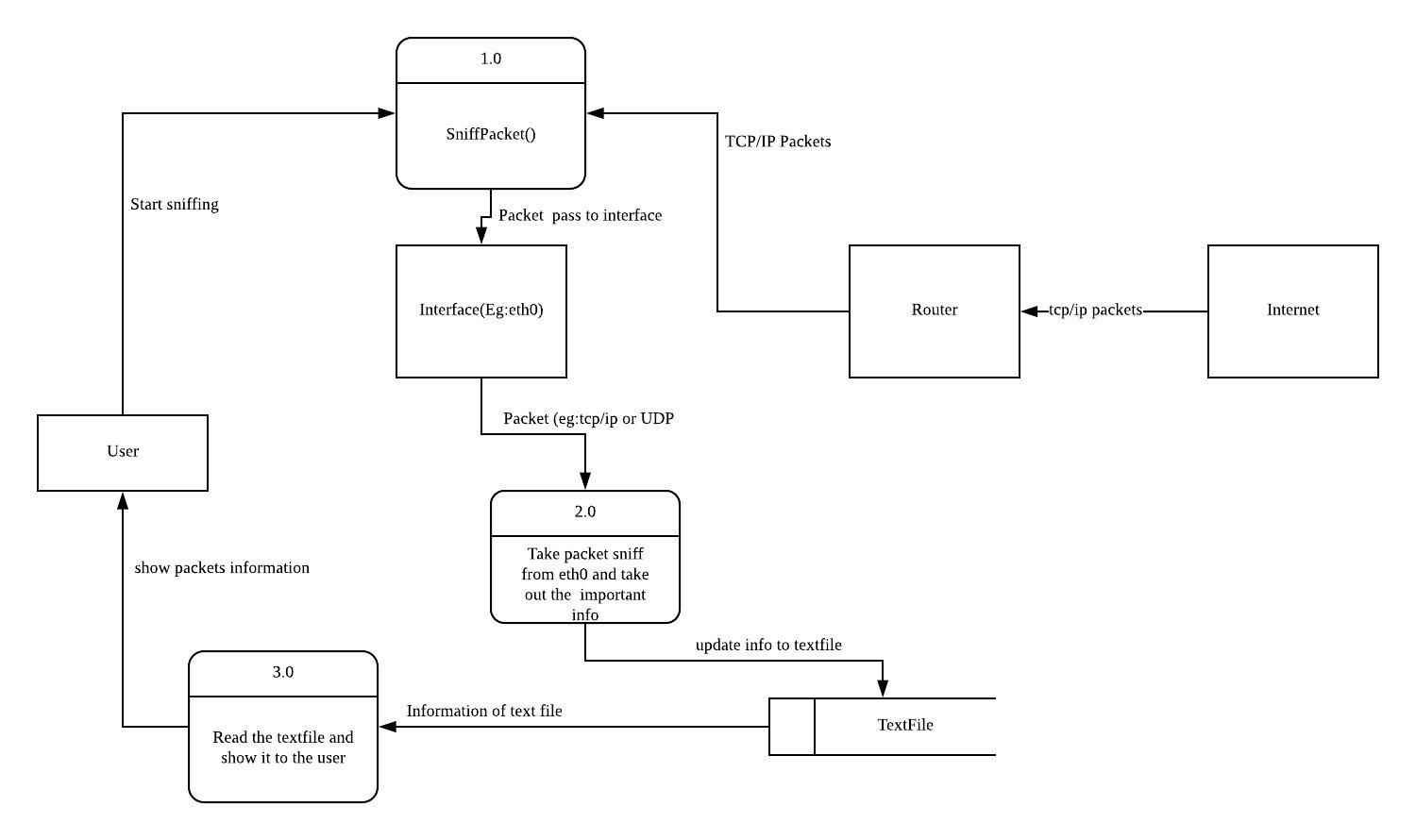


Pcbin activity diagram:



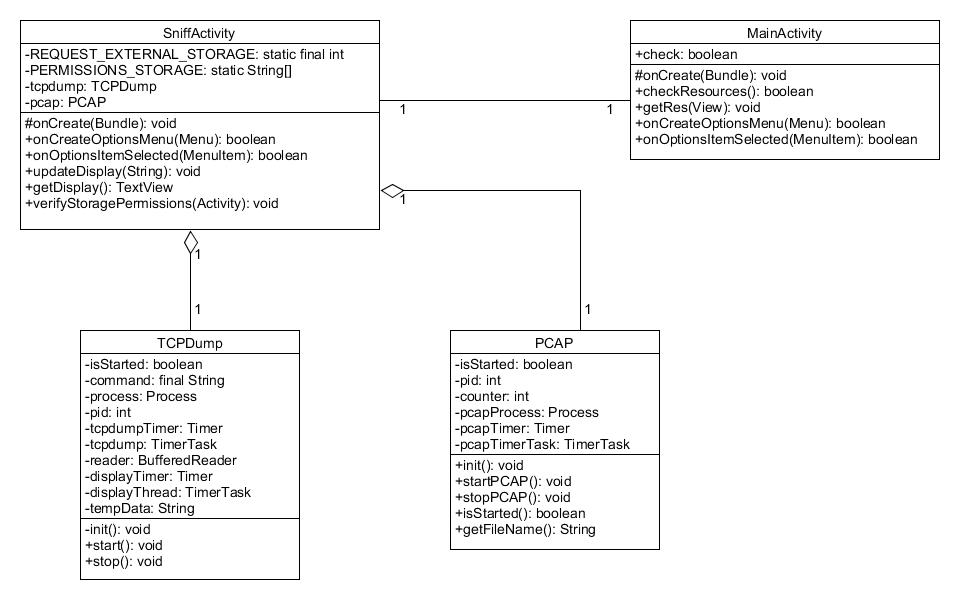
## Design Artefacts – Data Flow Diagram (Iteration 1)

Pcbin data flow diagram:



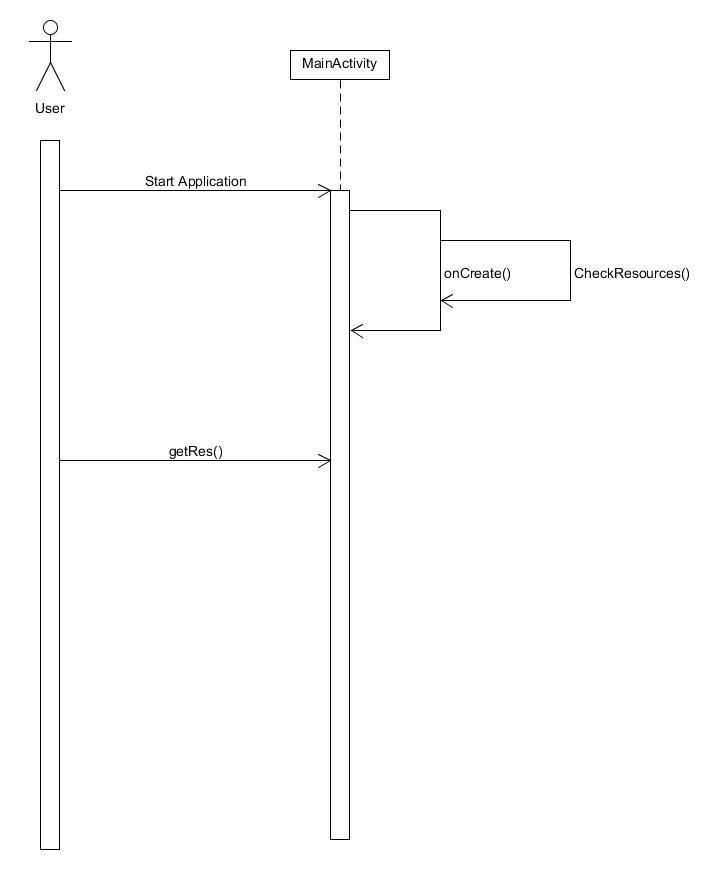
## Design Artefacts - Class Diagrams (Iteration 1)

Main class diagram:

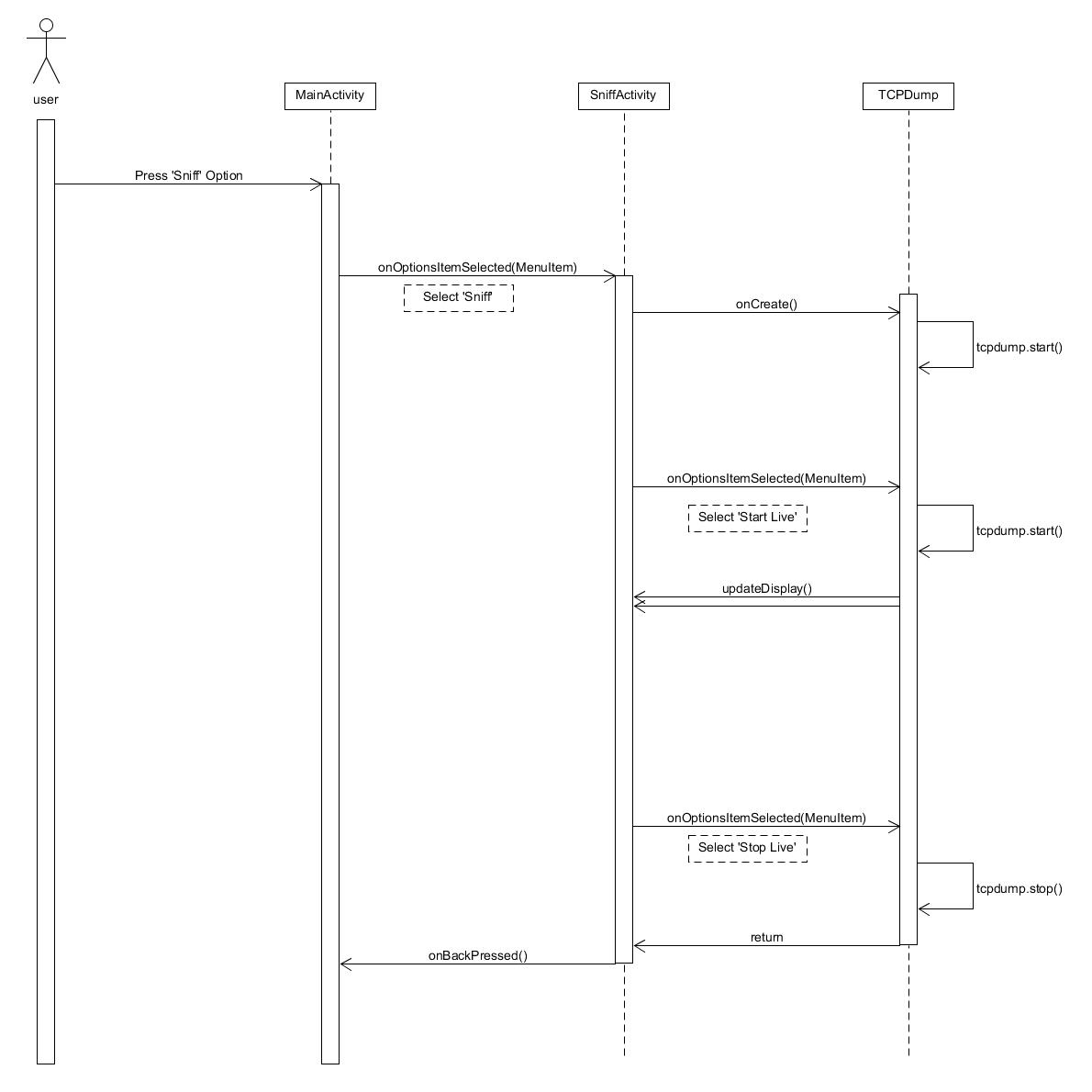


## Design Artefacts - Sequence Diagrams (Iteration 1)

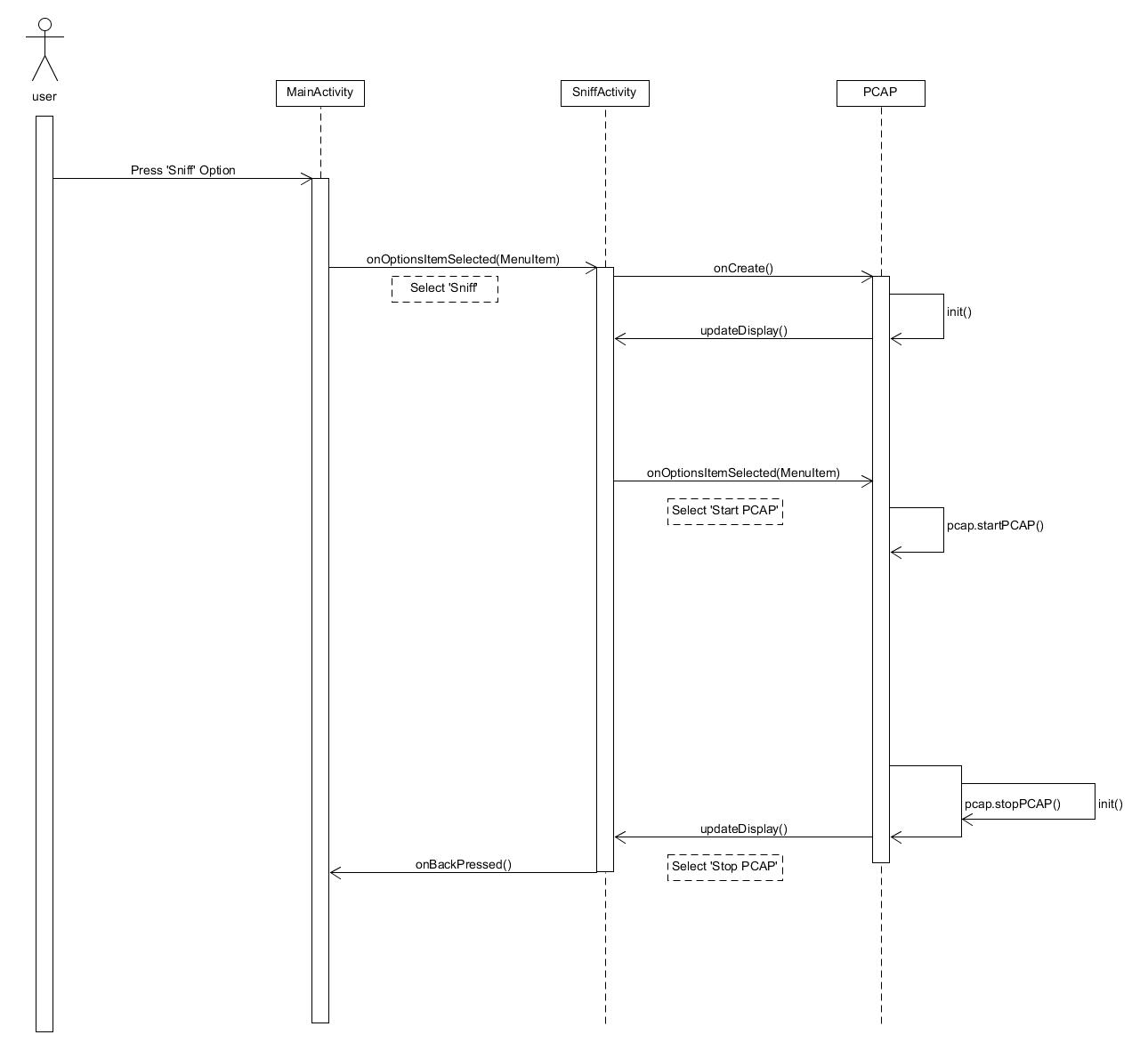
App initialize sequence diagram:



Start live sequence diagram:



Start pcap sequence diagram:



# 5. Test Plan Design

Depending on your system's software architecture, there may be many different classes, with many different implemented methods(), whose functionalities need to be tested.

Your team should describe how they intend to test their functionalities to ensure the methods() work as designed.

Below table provides a sample structure, to organize your content. You can reuse this structure to cover as many other important methods() as necessary.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Plan for :  Being able to capture/sniff Packets <Relate to fuctional requirement>Club Membership Management - The system should allow the administrator to add new members | | | | |
| Test Case ID: | Description | Expected Outcome | Participating Program's Class & Method to be tested | Participating CPPUnit Test Program's Class & Method doing the testing |
| E.g.  CM-1 | E.g.  To test for adding a new member's name "John Anderson" | E.g.  System should output a message like "Member : John Anderson created successfully".  The last array slot in the member array should contain the string " John Anderson " | E.g.  Class Name : ManageMember  Method Name :  bool createNewMember (string name) | E.g.  CPPUnit Class Name : TestManageMember  Method Name :  void testCreateNewMember (string name) |
| E.g.  CM-2 | E.g.  To test for adding an **EXISTING DUPLICATE** member's name "John Anderson" | E.g.  System should output a message like "Error, member : John Anderson already exists".  The total no. of elements in the member array should remain the same | E.g.  Class Name : ManageMember  Method Name :  bool createNewMember (string name) | E.g.  CPPUnit Class Name : TestManageMember  Method Name :  void testAddingExistingMember (string name) |
|  |  |  |  |  |
|  |  |  |  |  |

# 6. Evidence of using VCS (Version Control Software)

Version Control Software used : Github

Note 1 : The following screenshots can span as many pages as necessary

Note 2 : Please ensure that all the wordings / text in your screen shots are clearly visible

Screenshot #1 - Contents in the VCS's **Root Folder** containing all **Repository Project Files**

< Paste Screenshot #1 image here >

Screenshot #2 - VCS's listing of all the **latest source files currently being managed**

< Paste Screenshot #2 image here >

Screenshot #3 - Example using VCS's to **check-out source files** (it may be necessary to do >1 screen capture, depending on the software used)

< Paste Screenshot #3 image**(s)** here >

Screenshot #4 - Example using VCS's to **check-in source files** (it may be necessary to do >1 screen capture, depending on the software used)

< Paste Screenshot #4 image**(s)** here >

Screenshot #5 - Example using VCS's to **display the change history / log** (it may be necessary to do >1 screen capture, depending on the software used)

< Paste Screenshot #5 image**(s)** here >