Software Requirements Specification

For

Packet Sniffer

Version 1.0

Sunil Baliganahalli NarayanMurthy
Nehal Kamat
Apoorva Bapat

University of Colorado, Boulder

Mar 09, 2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Sunil Baliganahalli Narayana Murthy	2/17/2016	Initial draft	0.0
Sunil Baliganahalli Narayana Murthy	2/21/2016	Incorporated review comments from teammates	0.1
Sunil Baliganahalli Narayana Murthy	3/4/2016	Incorporated review comments from teammates	0.2
Apoorva Bapat	3/5/2016	Incorporated UI mockups	0.3
Nehal Kamat	3/6/2016	Incorporated updated use case	0.5
Apoorva Bapat	3/7/2016	Included Activity & Sequence diagrams	0.6
Nehal Kamat	3/7/2016	Included Activity & Sequence diagrams	0.7
Sunil Baliganahalli Narayana Murthy	3/7/2016	Included Activity & Sequence diagrams	1.0

1. Introduction

1.1 Purpose

Packet sniffing is defined as a technique that is used to monitor every packet that crosses the network. A packet sniffer is a piece of hardware or software that monitors all network traffic. Using the information captured by the packet sniffers an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help to maintain efficient network data transmission. For most organizations packet sniffer is largely an internal threat.

Packet sniffers can be operated in both switched and non-switched environment. Determination of packet sniffing in a non-switched environment is technologies that can be understand by everyone. In this technology all hosts are connected to a hub. There are a large number of commercial and non-commercial tools are available that makes possible eavesdropping of network traffic. Now a problem comes that how this network traffic can be eavesdrop; this problem can be solved by setting network card into a special “promiscuous mode”. Now businesses are updating their network infrastructure, replacing aging hubs with new switches. The replacement of hub with new switches that makes switched environment is widely used because “it increases security”. However, the thinking behind is somewhat flawed. It cannot be said that packet sniffing is not possible in switched environment. It is also possible in switched environment.

1.2 Intended Audience and Reading Suggestions

This document is intended for User, Developer and tester.

2. System Features

Business Requirements - [Not Applicable]

User Requirements				
ID	**Requirements**	**Topic Area**	**User**	**Priority**
UR-001	User should be able to launch application.	Interaction	Any	High
UR-002	User should be able to close the application	Freedom	Any	Medium
UR-003	User should be able to start capturing packets by selecting network interface		Any	High
UR-004	User should be able to stop capturing packets.		Any	High
UR-005	User should be able to mark packets for saving packet information.		Any	Medium
UR-006	User should be able to save either all the captured packets or marked captured packets.		Any	High
UR-007	User should be able to import/export the saved packets.		Any	High
UR-008	User should be able to view types of protocols used in captured packets.		Any	High
UR-009	Users should have the option of choosing the client machine to monitor packets from.	Freedom	Any	High
UR-010	User should be able to filter packets according to detected protocols.		Any	Medium
UR-011	User should be able to inspect the packet information for a selected packet		Any	Medium
UR-012	User should be able to view only packet header		Any	Medium
UR-013	User should have the option to view real time network statistics	Statistics	Any	High
UR-014	User should be able to validate a selected packet for its integrity and authenticity	Validation	Any	High

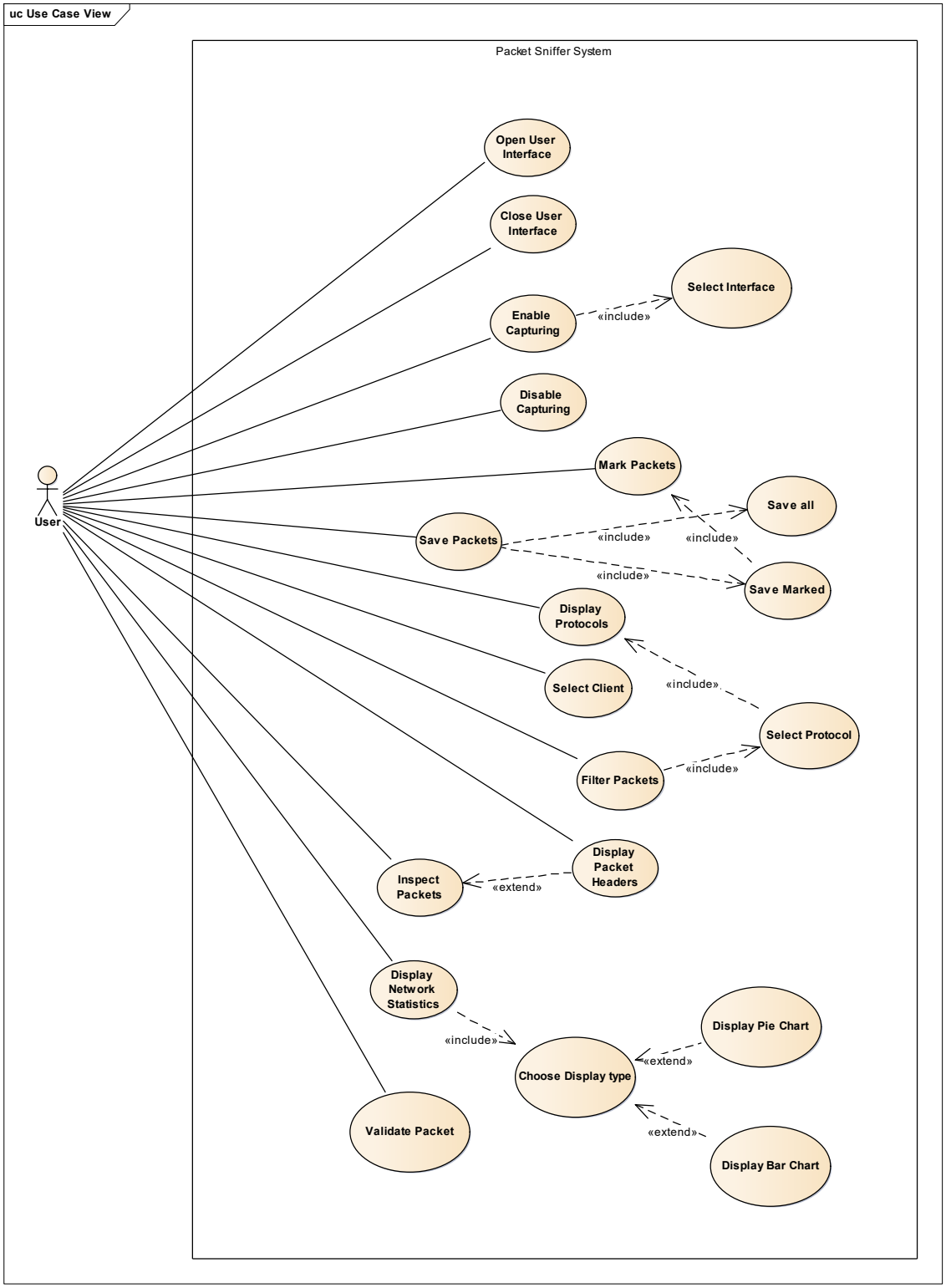
Functional Requirements				
ID	**Requirements**	**Topic Area**	**User**	**Priority**
FR-001	System should detect and display new clients in real time.		System	High
FR-002	System should include timestamp for the captured packets.		System	Medium
FR-003	System should remember user preference (export type, filter protocol) from last session.		System	Medium
FR-004	System should be able to summarize the statistics about the packets captured		System	Medium

Non-Functional Requirements				
ID	**Requirements**	**Topic Area**	**User**	**Priority**
NF-001	Application should work with sufficient network bandwidth.	Performance		High
NF-002	The application should be reliable.	Reliability		High
NF-003	Application should be robust and handle at-least 5 clients.	Scalability		High
NF-004	Application should be responsive.	Usability		High
NF-005	Application should respond to user action within 1sec.	Performance		Medium

3. Use Cases:

Actors: All users

Use Case Overview:



Use Case Documents:

Use Case ID:	UR-001
Use Case Name:	Open User Interface
Description:	Select application icon on desktop/ in the start menu to open a graphical interface for running the application

Actors:	Any		
Pre-conditions	User should choose to use graphical interface to application		
Post conditions	User should understand the layout of the interface and should understand how the information is being displayed		
Frequency of Use:	Moderate - High		
Flow of Events:		Actor Action	System Response
	1	Double-click application shortcut on desktop	Application GUI opens

Use Case ID:	UR-002
Use Case Name:	Close User Interface
Description:	Display the network statistics on the command line instead of a graphical interface

Actors:	Advanced Users		
Pre conditions	User should have application running		
Post conditions	User Interface closes		
Frequency of Use:	frequently		
Flow of Events:		Actor Action	System Response
	1	Close User Interface	Stop capturing packets. Close UI

Use Case ID:	UR-003
Use Case Name:	Enable Capturing
Description:	Allows the user to start capturing packets in the network

Actors:	All users		
Pre conditions	User should have		
Post conditions	Users should have opened either the graphical interface or command line interface		
Frequency of Use:	Frequently		
Flow of Events:		Actor Action	System Response
	1	Open application	Application user interface is displayed
	2	Click 'Enable Capturing'	Transmitted packet details are displayed on the UI

Use Case ID:	UR-004
Use Case Name:	Disable Capturing
Description:	Allows user to stop capturing packets in network

Actors:	All users		
Pre conditions	Application should be running and packets are being monitored		
Post conditions	Capturing of packets is stopped and user can use this data to analyze network		
Frequency of Use:	Very frequent		
Flow of Events:		Actor Action	System Response
	1	Start application	Application interface displayed to user
	2	Click Enable monitoring	Packets start being monitored and their information displayed on the interface
	3	Click Disable Capturing	Capturing of packets is stopped

Use Case ID:	UR-005
Use Case Name:	Mark Packets
Description:	Enables the user to mark specific packets for saving information

Actors:	All users		
Pre conditions	Application should be running and packets being captured		
Post conditions	Packets are marked as per user's requirements for saving		
Frequency of Use:	Very frequent		
Flow of Events:		Actor Action	System Response
	1	Start application	Application interface displayed to user
	2	Click Enable Capturing	Packets start being monitored and their information displayed on the interface
	3	Select packet information to be saved by clicking check boxes against the packet names	Packet information is saved in a log file created in a pre-specified local directory

Use Case ID:	UR-006
Use Case Name:	Save Packets
Description:	Enables the user to save packet information

Actors:	All users		
Pre conditions	Application should be running and packets being monitored		
Post conditions	Packets information is saved according to user preference: either all packets are saved or only marked packets are saved.		
Frequency of Use:	Very frequent		
Primary Flow of Events:		Actor Action	System Response
	1	Click on file menu	Display file menu
	2	Click on save packets	Display submenu giving user the choice of saving either all or only marked packets
	3	Select ‘Save marked packets’	Display checkboxes in front of packets for marking packets for saving. Open dialogue for user to select export type
	4	Select text/xml/p-cap export format	Save packet information in user chosen format.
Alternative Flow of Events:	1	Click on file menu	Display file menu
	2	Click on save packets	Display submenu giving user the choice of saving either all or only marked packets
	3	Select ‘Save all packets’	Open dialogue for user to select export type
	4	Select text/xml/p-cap export format	Save packet information in user chosen format.

Use Case ID:	UR-007
Use Case Name:	Display packet protocols
Description:	Gives user the list different protocols used in captured packets.

Actors:	All users		
Pre conditions	Users should start the application and click on display packet protocols.		
Post conditions	Users should be displayed a list of all protocols used in the captured packets		
Frequency of Use:	High		
Flow of Events:		Actor Action	System Response
	1	Click on View menu	Display View Menu
	2	Click Display packet protocols	A list of all protocols used in the captured packets

Use Case ID:	UR-008
Use Case Name:	Select Client
Description:	User is able to select a client to capture packets

Actors:	All users		
Pre conditions	Users should start the application.		
Post conditions	User should be able to see packets captured only from selected clients		
Frequency of Use:	High		
Flow of Events:		Actor Action	System Response
	1	Start the application	User interface displayed
	2	Select client from a drop down list	Packets only from selected client are displayed

Use Case ID:	UR-009
Use Case Name:	Filter packets
Description:	User should be able to display packets having a specific protocol

Actors:	All users		
Pre conditions	Users should start the application and select the type of packets of their preference		
Post conditions	Users should be displayed only those type of packets that have been filtered out by the user		
Frequency of Use:	High		
Flow of Events:		Actor Action	System Response
	1	Start application	Open User Interface
	2	Click Filter Packets	Display List of Protocols
	3	Double Click Protocol	Set condition to display packets with selected protocol only
		-	Start Capturing Packets

Use Case ID:	UR-010
Use Case Name:	Inspect Packets
Description:	Enable users to view packet info of selected packet

Actors:	All users		
Pre conditions	Users should start the applications and start capturing packets		
Post conditions	User should be displayed packet information		
Frequency of Use:	Low		
Flow of Events:		Actor Action	System Response
	1	Start application	Open User Interface
	2	Enable Capture Packets	Display captured packets
	3	Select Packet	-
	4	Click Inspect Packet	Display all packet information

Use Case ID:	UR-011
Use Case Name:	Display Packet Header
Description:	Enables users to view only header of selected packet

Actors:	All users		
Pre conditions	Users should start the application, start monitoring packets and select the packet whose header is to be expanded		
Post conditions	Users should be displayed the entire packet header		
Frequency of Use:	Low		
Flow of Events:		Actor Action	System Response
	1	Start application	Open User Interface
	2	Enable Capture Packets	Display captured packets
	3	Select Packet	-
	4	Click Inspect Packet	Display all packet information
	4	Right click packet	-
	5	Select Display Packet Header	Display only packet header

Use Case ID:	UR-012
Use Case Name:	Display Network Statistics
Description:	Enables user to view real time statistics of the information being transmitted along the network

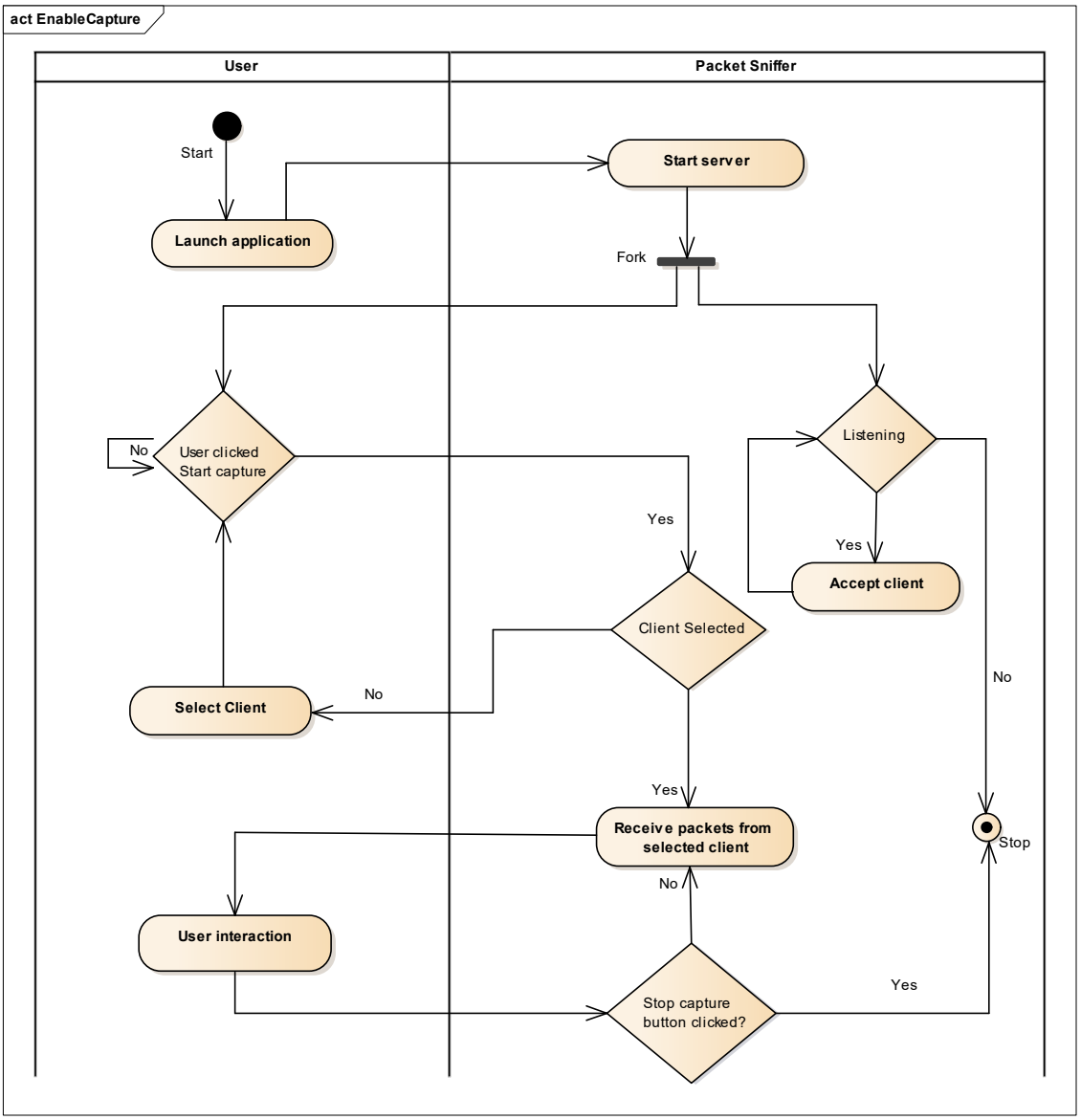
Actors:	All users		
Pre conditions	Users should start the applications and start capturing packets		
Post conditions	Users should be displayed real-time statistics of packets in the form of pie charts or bar graphs		
Frequency of Use:	High		
Flow of Events:		Actor Action	System Response
	1	Start application	Open User Interface
	2	Enable Capture Packets	Display Captured Packets
	3	Click Display Network Statistics	Display chooser with 2 options – Bar Graph and Pie Chart
	4	Select Bar Graph/Pie Chart	Display Appropriate Plot

Use Case ID:	UR-013
Use Case Name:	Validate Packet
Description:	Enables user to check authenticity of the packet and whether or not it is broken

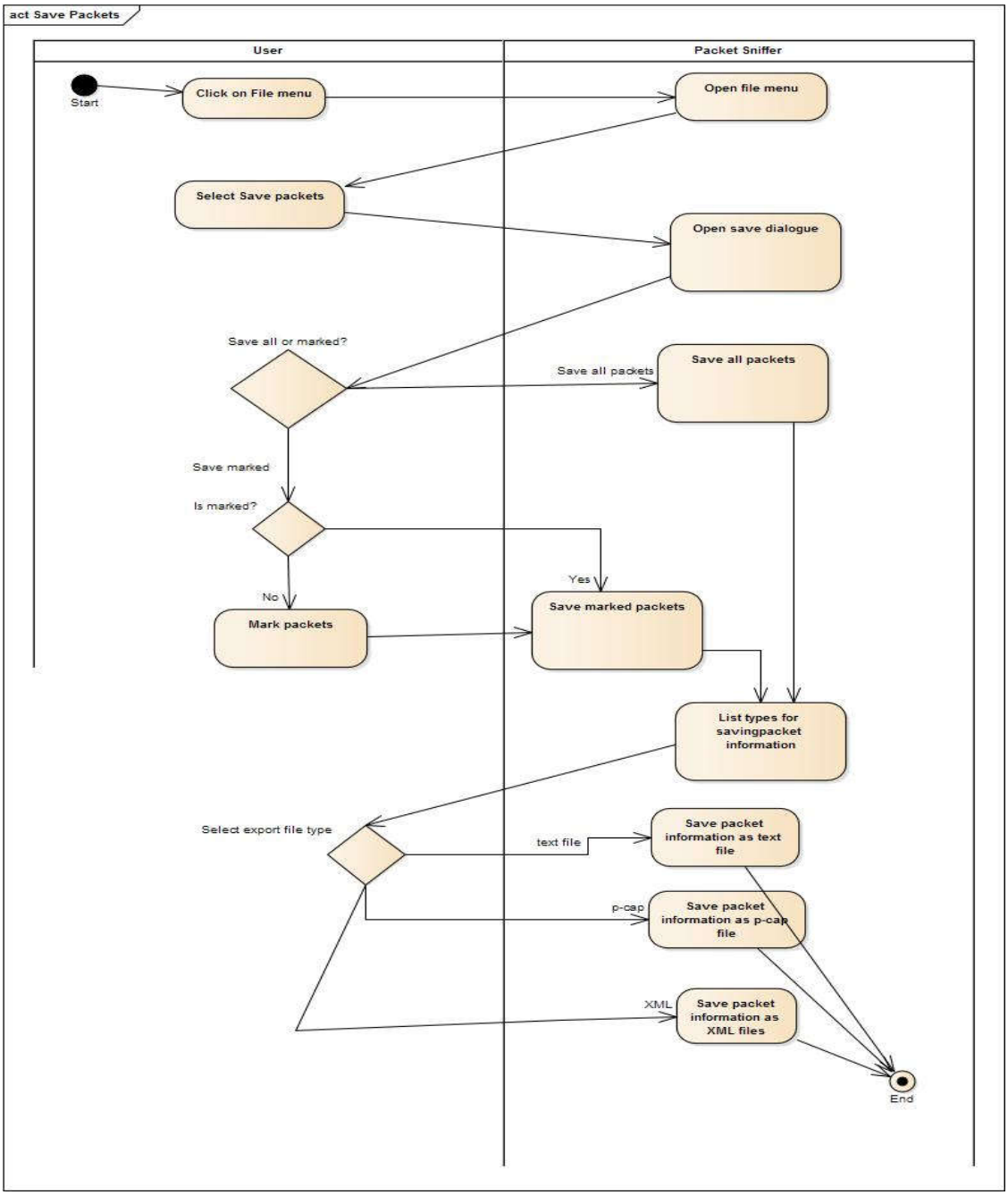
Actors:	All users		
Pre conditions	Users should start the applications and start capturing packets		
Post conditions	User should be displayed with information regarding the checksum of the packet and whether or not it is broken/incomplete		
Frequency of Use:	Moderate		
Flow of Events:		Actor Action	System Response
	1	Start application	Open User Interface
	2	Enable Capture Packets	Display Captured Packets
	3	Select Packet	-
	4	Click Validate Packet	Display checksum and packet integrity information

4. Activity Diagrams:

+ Requirement ID : {UR-003} | Use Case ID : {UR-003} | Use Case Name : {Start Capture} | Group Member Name : Sunil Baliganahalli Naryana Murthy



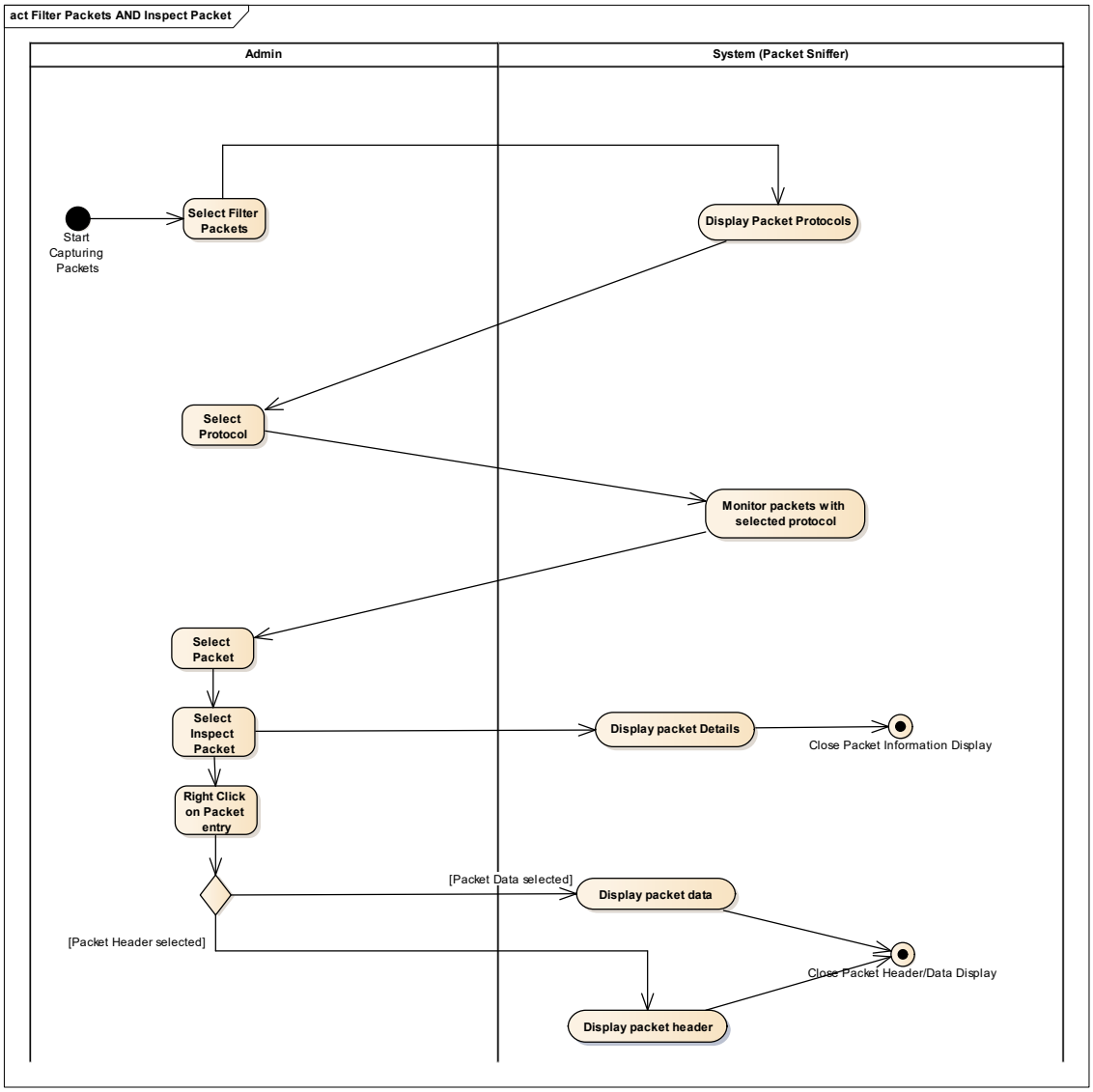
+ Requirement ID : {UR-006} | Use Case ID : {UR-006} | Use Case Name : {Save Packets} | Group Member Name : Apoorva Bapat



Use Case Name: {Save Packets}

Description: This diagram represents the activity of saving packets according to user preference of saving only the marked packets. The user clicks on file menu and then UI prompts the user to fill in its requirements. Once, the system gets its marked packets, it then prompts user to select file type or exporting and saves in the selected format.

+ Requirement ID : {UR-009, UR-010} | Use Case ID : { UR-009, UR-010} | Use Case Name : {Filter Packets, Inspect Packet} | Group Member Name : Nehal Kamat
[Note : 2 use cases combined]



Use Case Name: {Filter Packet, Inspect Packet}

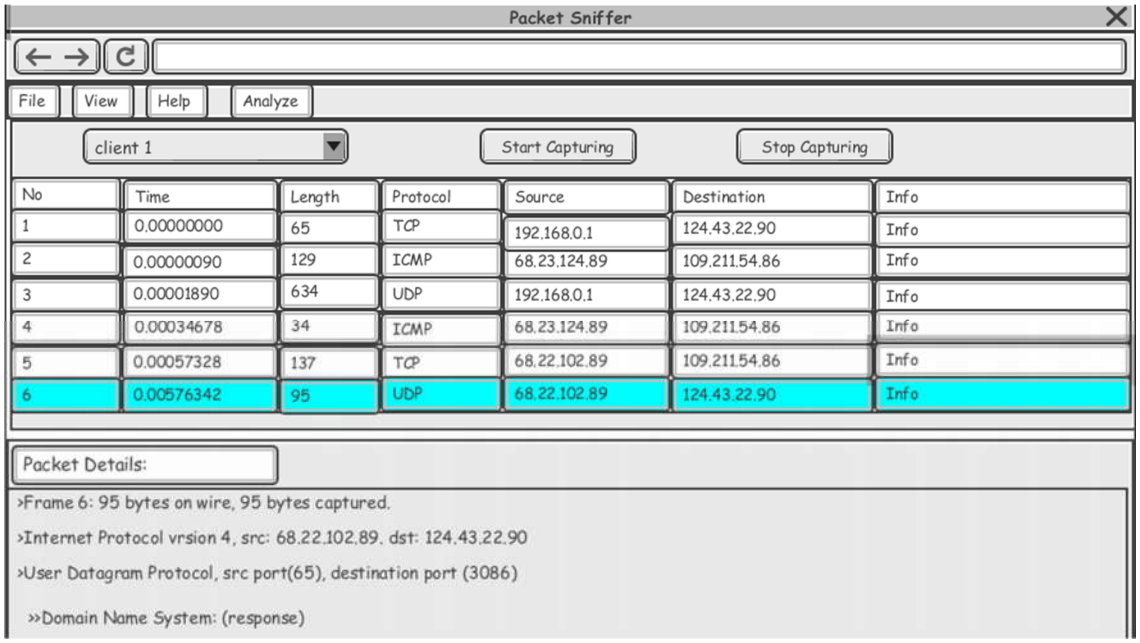
Description: This diagram represents the activity of the user filtering packets by protocol and then inspecting one packet from the filtered packets. The user first clicks filter packets which then gives the user the option of choosing one of the packet protocols. After choosing the protocol and the sniffer applying the filter to the captured packets, the user can select a packet and inspect its data.

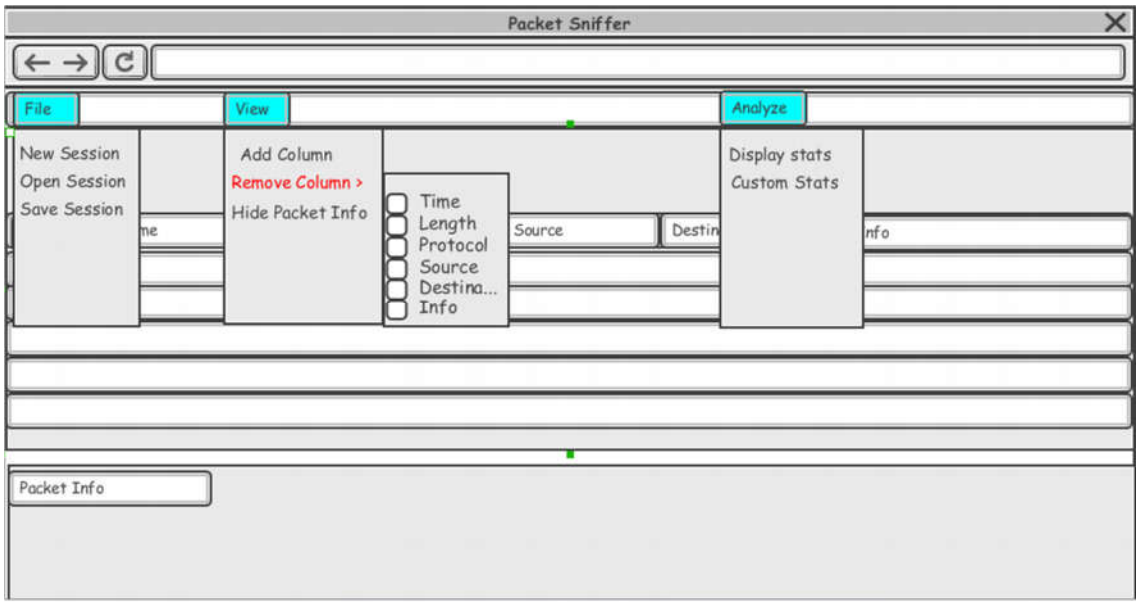
5 Data Storage: PCAP format, Text format, Xml format

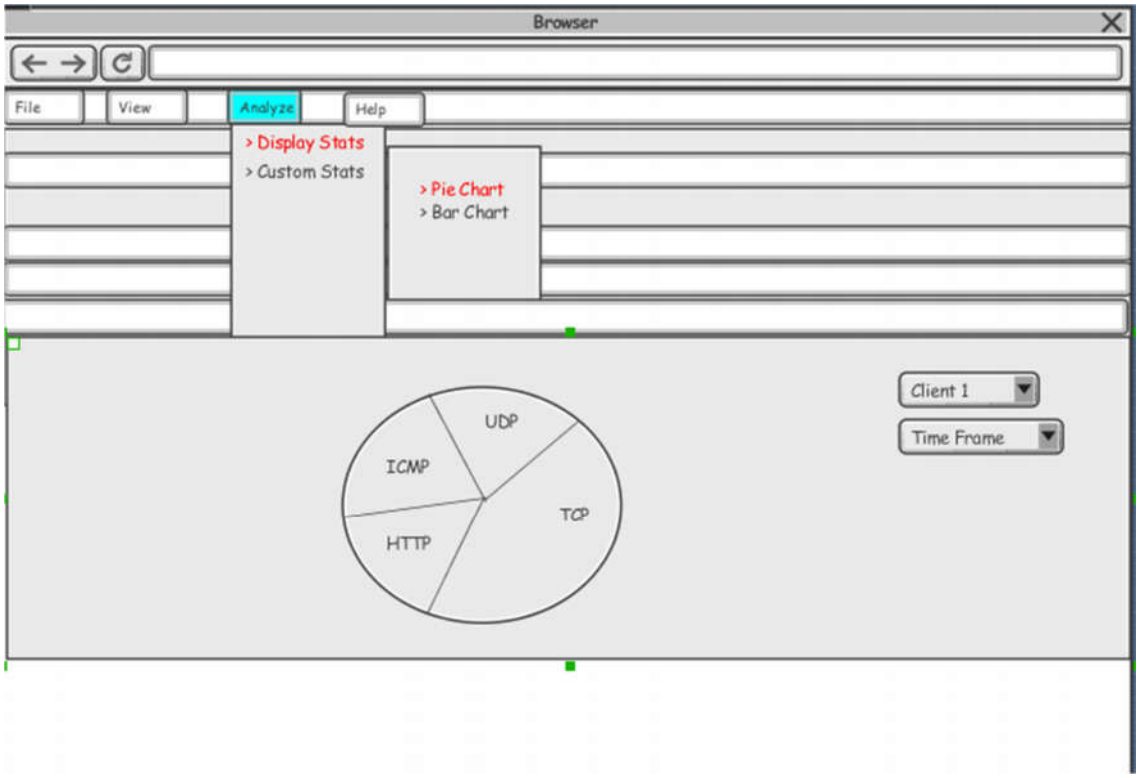
Classes:

- The application supports multiple formats like Pcap, Text, xml etc.
- All of these format implement a ImportExportData interface.
- The PcapImportExport supports impoting from Pcap and exporting in a Pcap format. Likewise for TextImportExportData, XmlImportExportData (shown in class diagram).

6 UI Mockups:

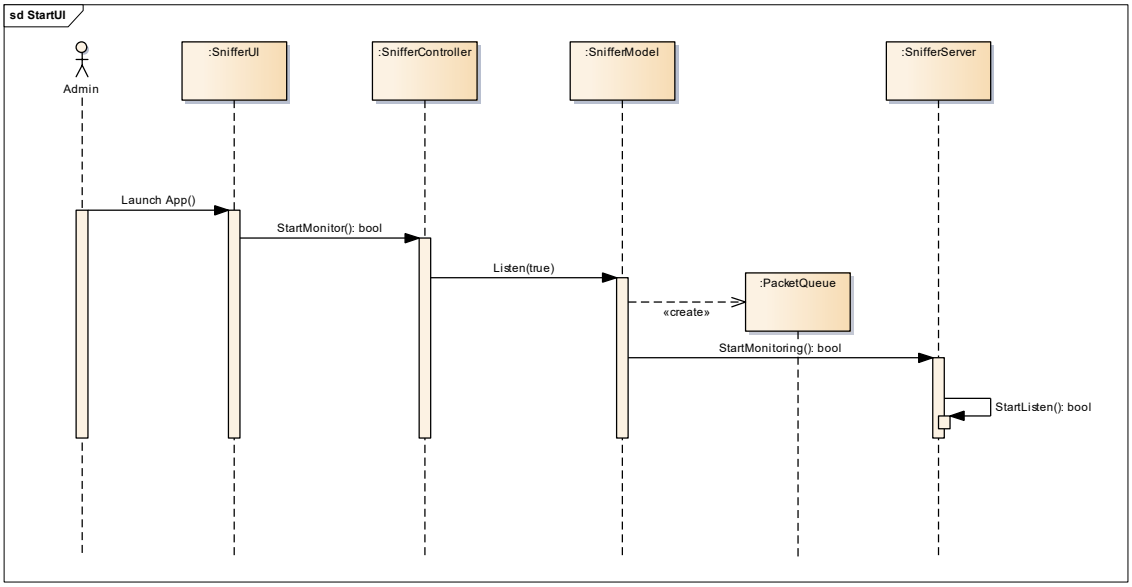



Packet Sniffer
File
View
Analyze
New Session
Open Session
Save Session
Add Column
Remove Column >
Hide Packet Info
Time
Length
Protocol
Source
Destina...
Info
Source
Display stats
Custom Stats
Packet Info


Browser
File
View
Analyze
Help
> Display Stats
> Custom Stats
> Pie Chart
> Bar Chart
UDP
ICMP
TCP
HTTP
Client 1
Time Frame

7 User Interactions:

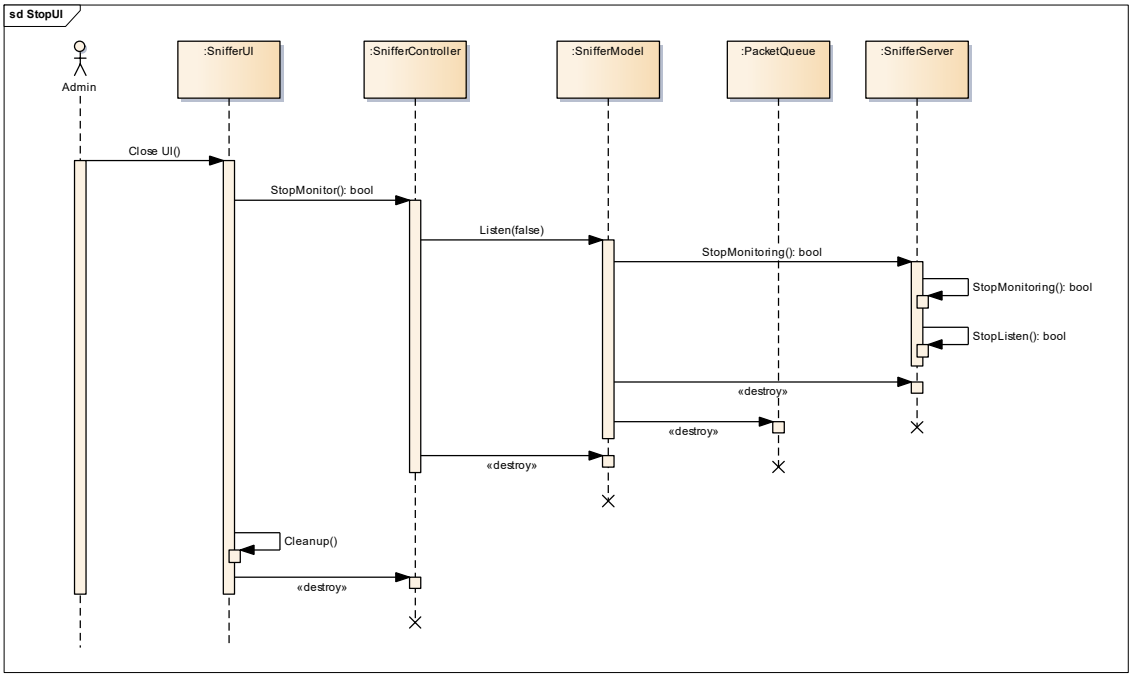
+ Requirement ID : {UR-001} | Use Case ID : {UR-001} | Use Case Name : {Start User Interface} | Group Member Name : Sunil Baliganahalli Naryana Murthy



Use case Name: Start user Interface

Description: The above sequence diagram shows the application boot-up sequence and different classes that are created thereafter.

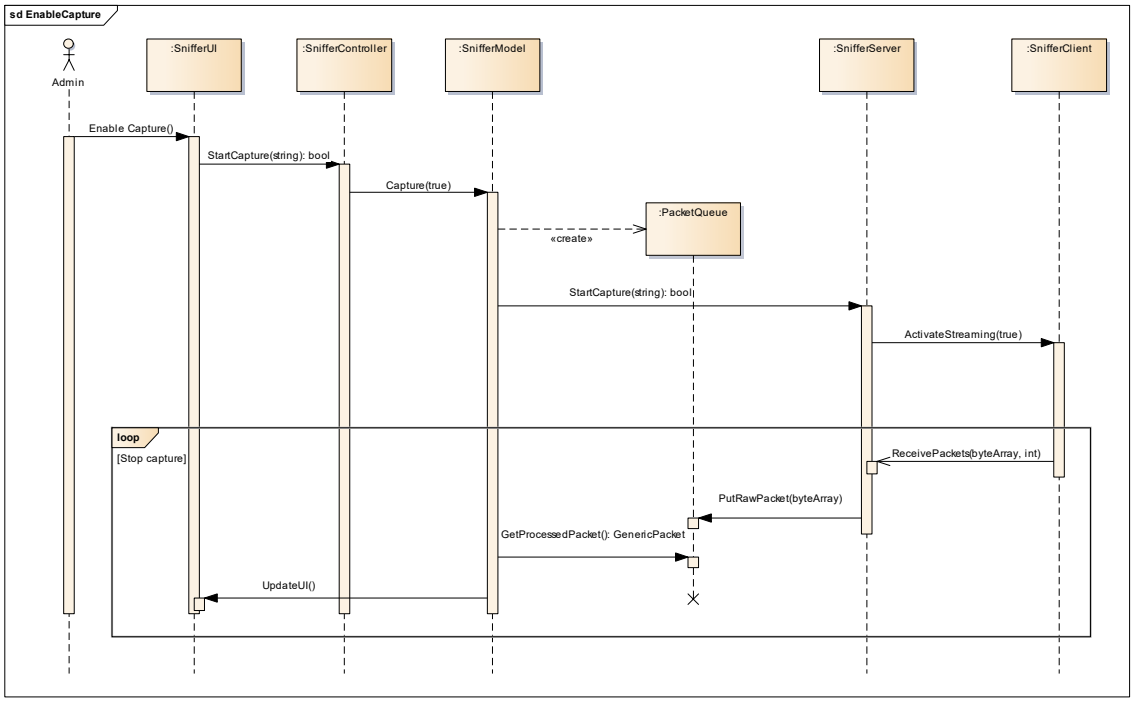
+ Requirement ID : {UR-002} | Use Case ID : {UR-002} | Use Case Name : {Close User Interface} | Group Member Name : Sunil Baliganahalli Naryana Murthy



Use case Name: Close user Interface

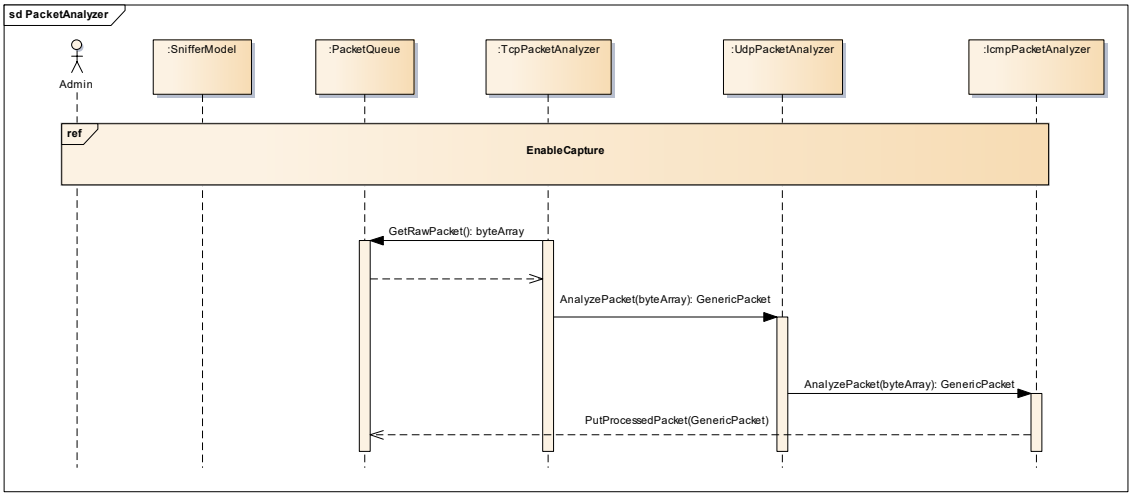
Description: The above sequence diagram shows the application shutdown sequence and different classes that are destroyed thereafter.

+ <u>Requirement ID</u> : {UR-003} | <u>Use Case ID</u> : {UR-003} | <u>Use Case Name</u> : {Enable Capturing} | <u>Group Member Name</u> : Sunil Baliganahalli Naryana Murthy

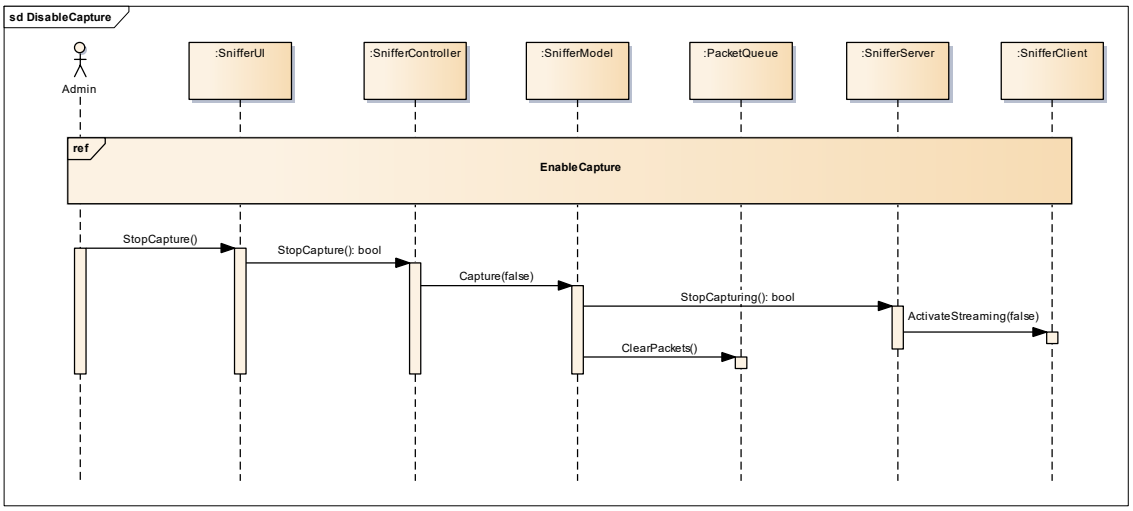


Use case Name: Enable capturing

Description: The start capture sequence starts with the user clicking on the enable capture. The packet sniffer server messages the sniffer server to start the listening to the incoming clients. Server then accepts any incoming client and adds the received packets to packet queue, which is then picked up by the packet analyzer to convert the byte stream of packets into object (Generic Packet).


sd PacketAnalyzer
Admin
:SnifferModel
:PacketQueue
:TcpPacketAnalyzer
:UdpPacketAnalyzer
:IcmpPacketAnalyzer
ref
EnableCapture
GetRawPacket(): byteArray
AnalyzePacket(byteArray): GenericPacket
AnalyzePacket(byteArray): GenericPacket
PutProcessedPacket(GenericPacket)

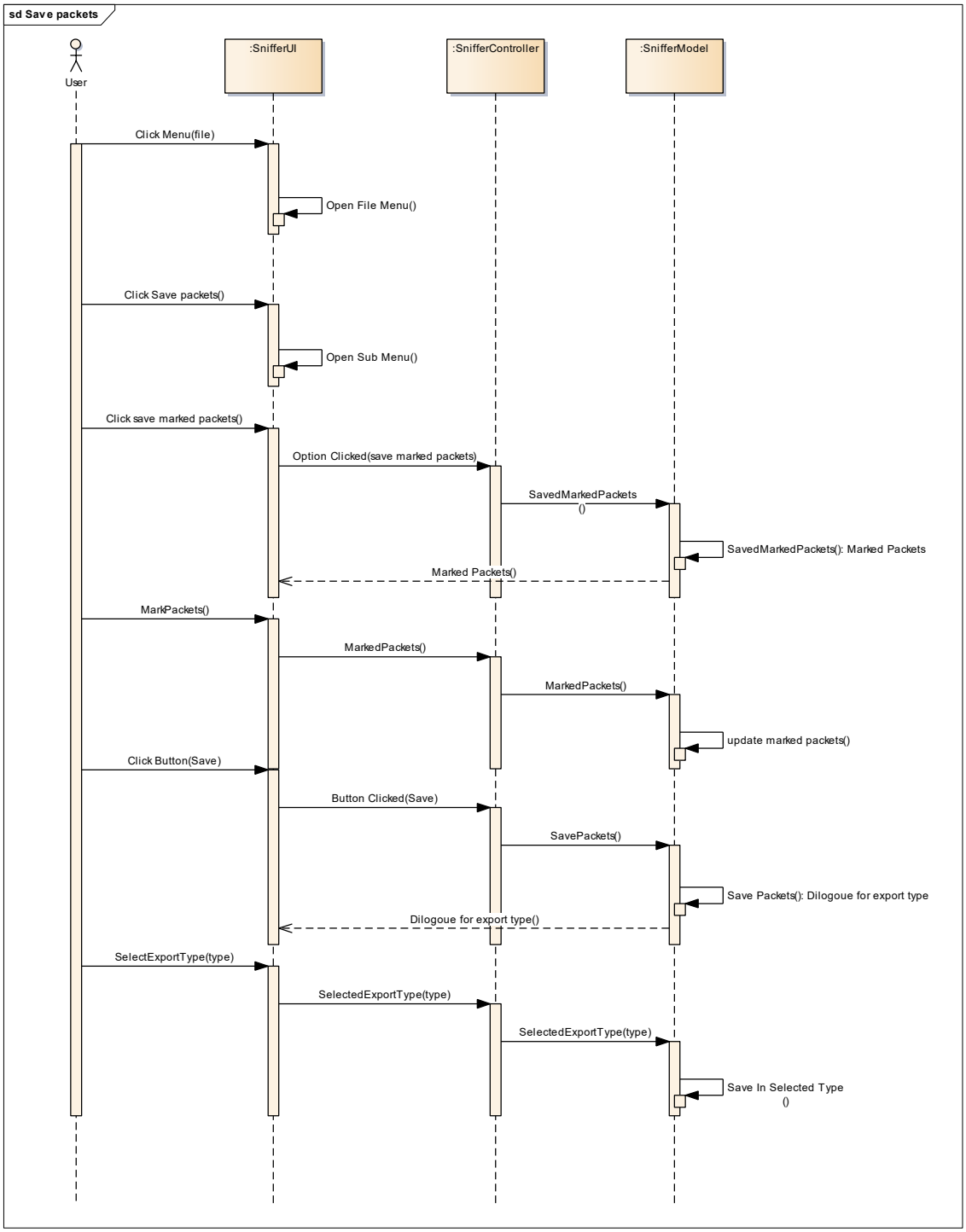
+ Requirement ID : {UR-004} | Use Case ID : {UR-004} | Use Case Name : {Disable Capturing} | Group Member Name : Sunil Baliganahalli Naryana Murthy



Use Case Name: Disable Capturing

Description: The stop sequence of the packet sniffer is shown below. The user clicks on disable capture which signals the sniffer model to stop streaming to the packets.

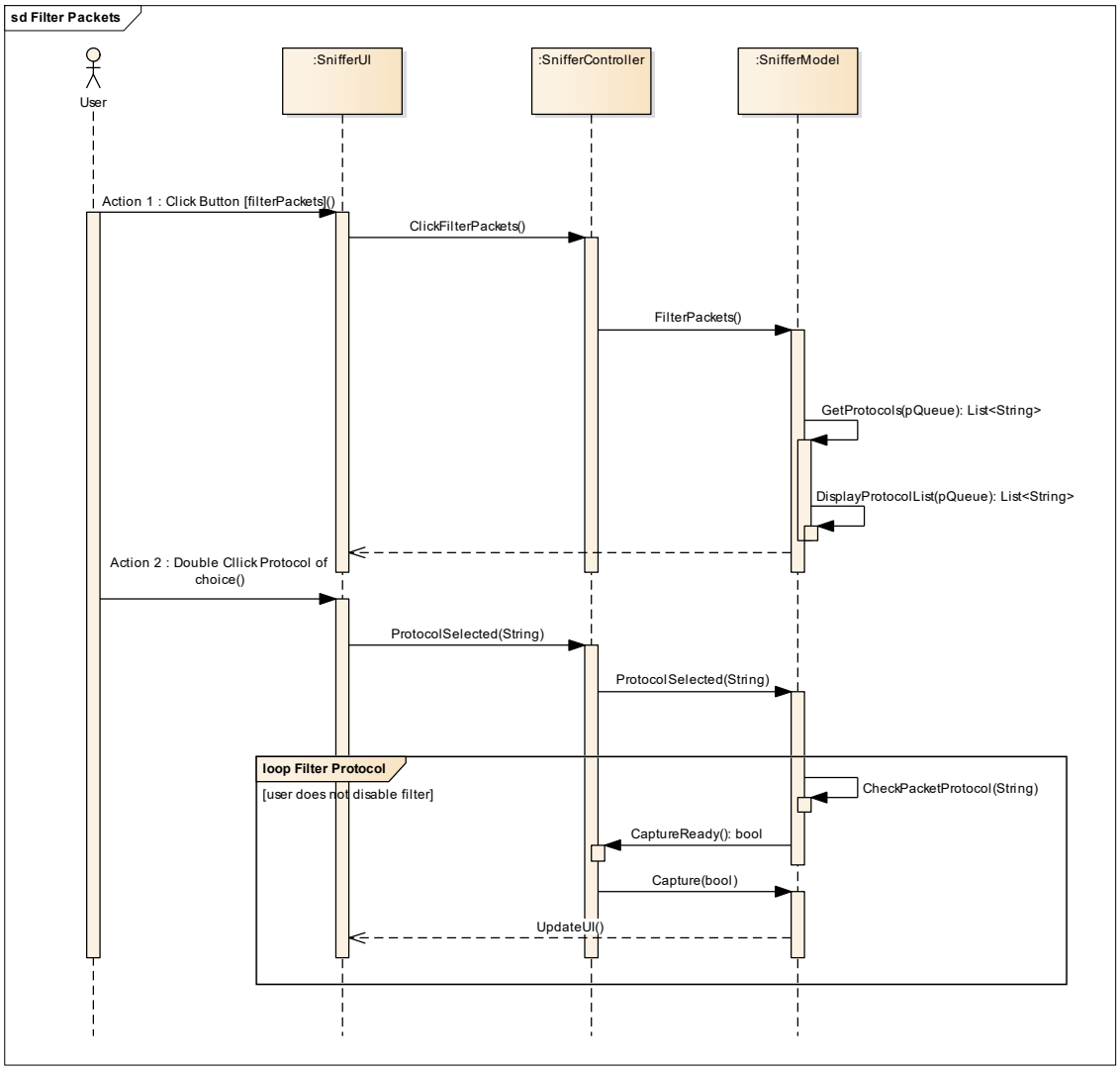
Requirement ID: {UR-006}, Use Case ID: {UC-006}, Use Case Name: {Save Packets}, Group Member Name: Apoorva Bapat



Use Case Name: Save Packets

Description: User clicks on File menu, which then leads to sub menus and selects save packets. User selects save marked packets for which he/she ahs to mark packets which are to be saved. After marking these packets, user selects the file format for exporting files.

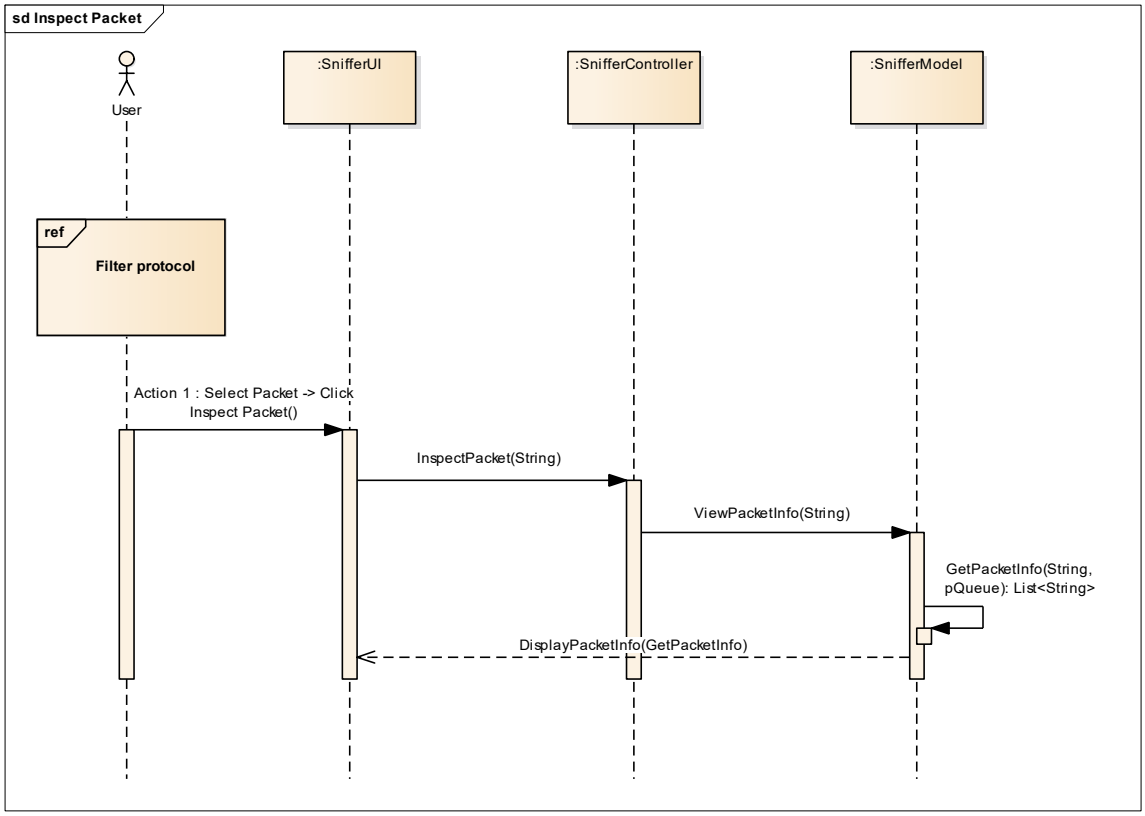
+ Requirement ID : {UR-009} | Use Case ID : { UR-009} | Use Case Name : {Filter Packets} | Group Member Name : Nehal Kamat



Use case Name: {Filter Packet}

Description: User clicks filter packet, which the system then processes to display the list of protocols. The user chooses the protocol and the system then filters the packet according to the chosen protocol. During the time when the filter is applied, the user can select and inspect a packet from the captured packets, which is described in the next sequence diagram.

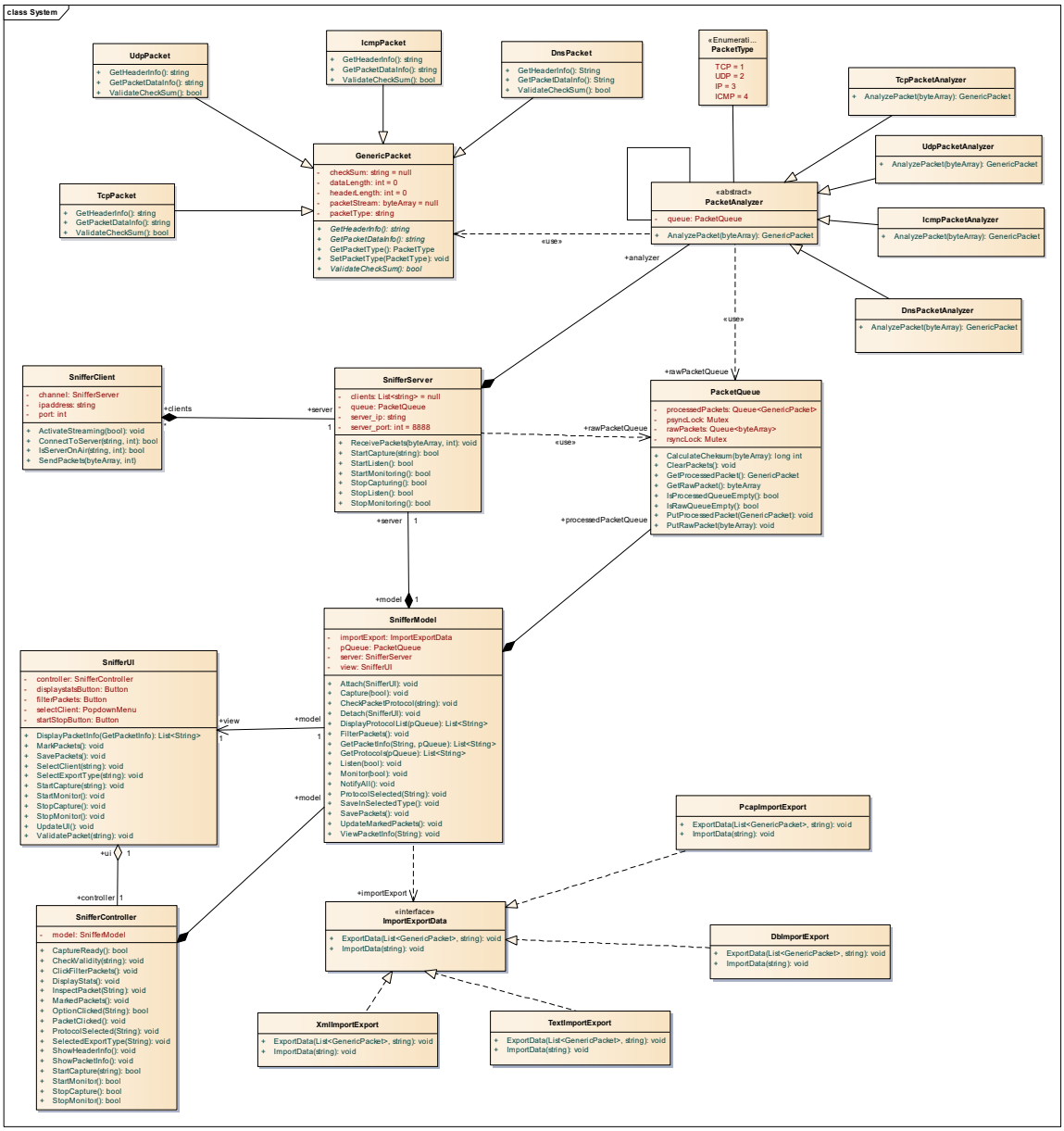
+ Requirement ID : {UR-010} | Use Case ID : { UR-010} | Use Case Name : {Inspect Packet} | Group Member Name : Nehal Kamat



Use case Name:
{Inspect Packet}

Description: This sequence of actions happens when a filter has been applied by the user. The user selects the packet and clicks inspect packet which the system then processes to display all the details of the packet.

7. Class Diagram:



The packet sniffer is client-server architecture. We are using Model-View-Controller (MVC) for the User interface and interfacing with the backend. The packet analyzer uses a Chain of responsibility pattern for analyzing the packet, which gives you the flexibility of extending the analyzer for other types of packets later. For import and export we use a strategy pattern which supports multiple import/exports formats like PCAP, XML etc.

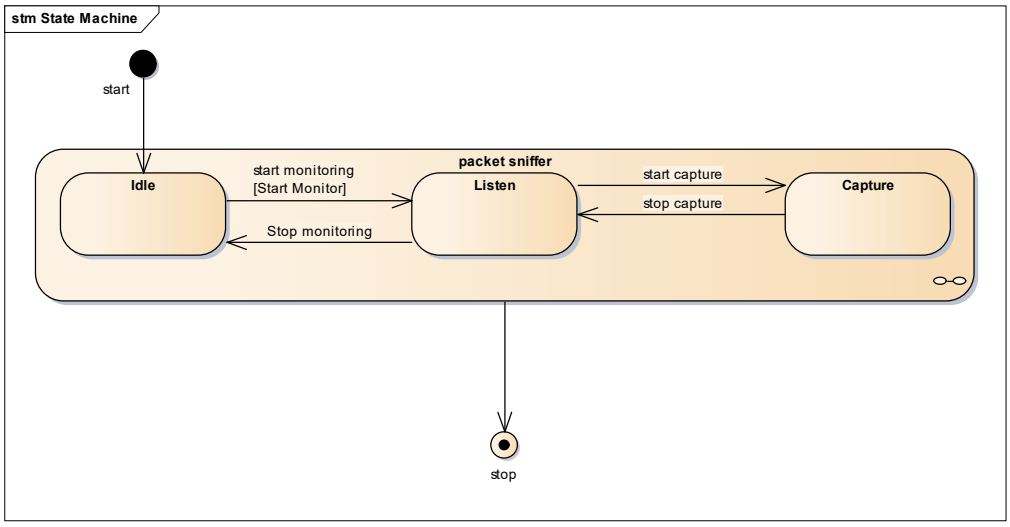
Architectural Pattern:

- Client-Server

Design Patterns used:

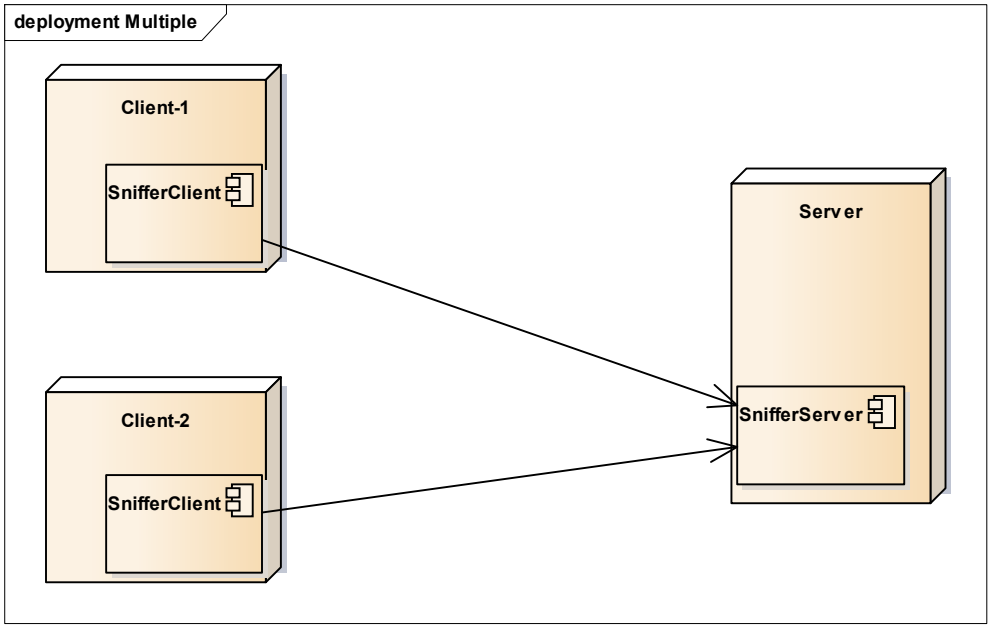
- Model-View-Model
- Observer pattern
- Strategy Pattern
- Chain of responsibility

8. State Machine Diagram:



9. Deployment View:

(i) Multi-client deployment:



(ii) Standalone deployment: