

# Dokumentation Projekt Netzwerksniffing

Ursprüngliche Autoren: Grigory Pavlov, Colin van Loo  
Datum der Erstellung: 11.09.2020  
Letzter Autor: Grigory Pavlov  
Datum der letzten Bearbeitung: 07.02.2021

## Inhaltsverzeichnis

Inhaltsverzeichnis.....	1
Änderungskontrolle.....	2
Dokumentspezifikation.....	3
Projektbeschreibung .....	3
Funktionale Anforderungen.....	3
Nicht-Funktionale Anforderungen.....	3
Klassendiagramm .....	4
Use-Cases.....	7
Sequenzdiagramme .....	8
GUI-Mockups.....	9
Main Menu – Add a listener.....	9
MainMenu – Listener .....	10
Reflexion.....	11
Abbildungsverzeichnis.....	12

## Änderungskontrolle

Die Version der Software entspricht der Version des Pflichtenhefts.

Version	Beschreibung	Verfasser	Datum
0.1	Entwurf des Pflichtenhefts	Grigory Pavlov	11.09.2020
0.2	Überarbeitung	Colin van Loo	18.09.2020
0.3	Anpassung des Pflichtenhefts an die neue Version mit VPN.	Colin van Loo	30.01.2020
1.0	Endversion Pflichtenheft	Grigory Pavlov	07.02.2021

## Dokumentspezifikation

In diesem Dokument sind die Ursprüngliche Idee, die Bestandteile des Programms, allfällige längere Kommentare und Beschrieb der Funktionen des Projekts ausführlich erklärt.

## Projektbeschreibung

Es wird eine Applikation zum Anzeigen von Netzwerkpaketen programmiert. Es sollen Pakete angezeigt, abgefangen und manipuliert werden können. Das Programm wird in Android Studio mit Java programmiert und soll auf Android laufen.

## Funktionale Anforderungen

Beschreibung	Muss-Kriterium	Kann-Kriterium
Anzeigen von übers Netzwerk gesendeten Pakete.	x	
Abfangen dieser Pakete	x	
Manipulieren dieser Pakete		x

## Nicht-Funktionale Anforderungen

Es sollen zum Projekt UML-Diagramme erstellt werden. Das Projekt muss innerhalb dieses Modules (226a) nicht fertig gestellt werden, sondern es kann im Modul 226b weitergearbeitet werden.

## Klassendiagramm

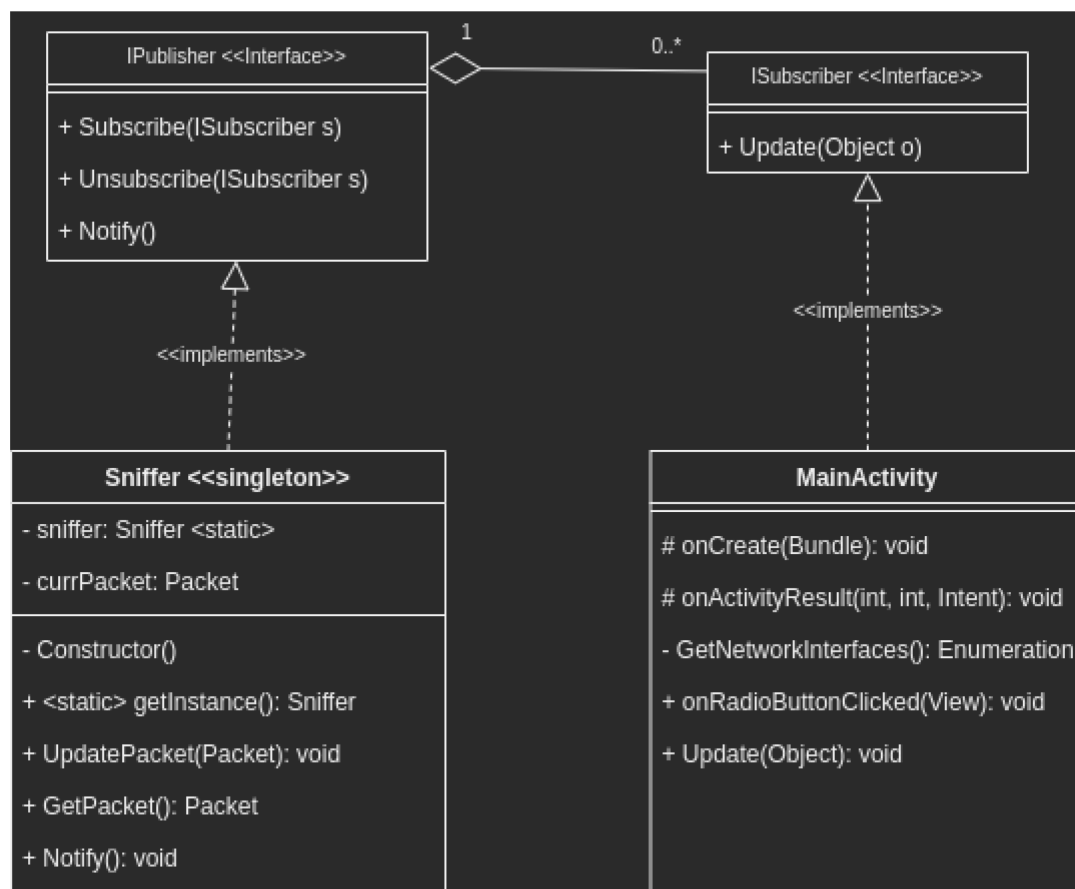


Abbildung 1: Klassendiagramm Netzwerksniffer

Klasse	Beschreibung
IPublisher	Interface fuer das observer-pattern
ISubscriber	Interface fuer das observer-pattern
Sniffer	Empfaengt die packets from vpn und sendet sie an alle Subscriber. Ist eine Singleton-Klasse.
MainAcitvity	Main-entry-point der Applikation

## Aufbau eines Packets

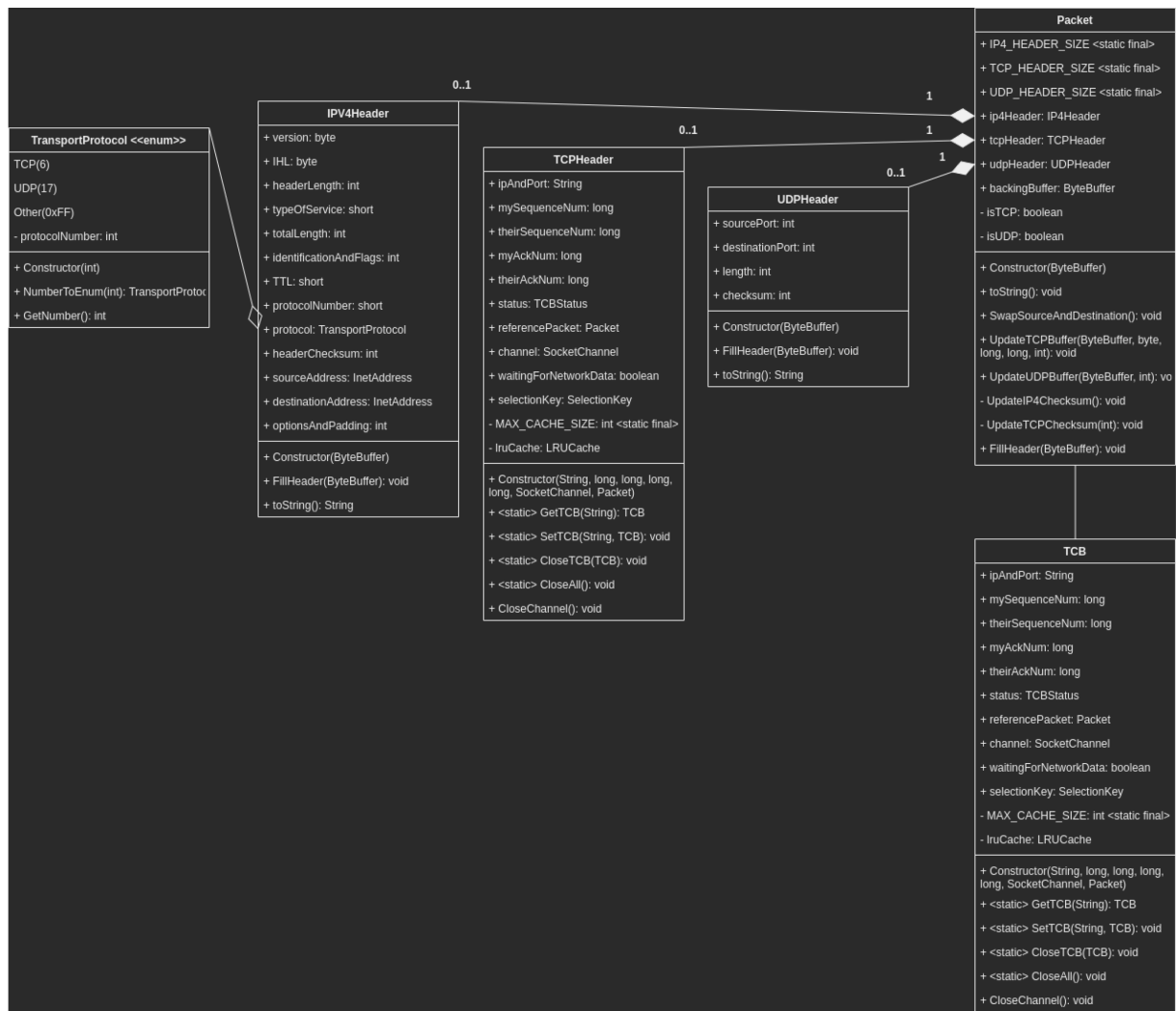


Abbildung 2: Grafische Darstellung des Aufbaus eines geschnittenen Pakets

## Aufbau des VPN-Service

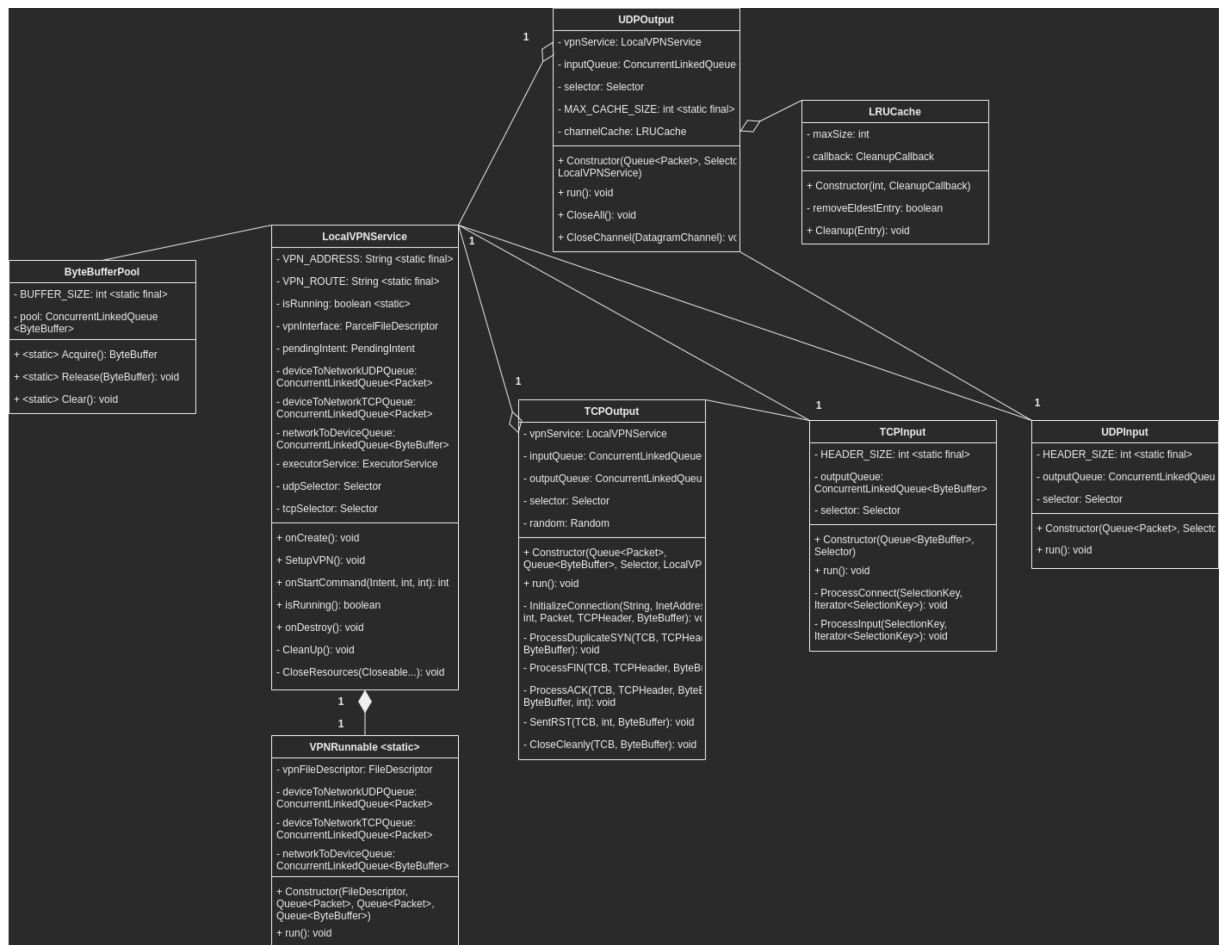


Abbildung 3: Grafische Darstellung des Aufbaus des VPN-Services

## Use-Cases

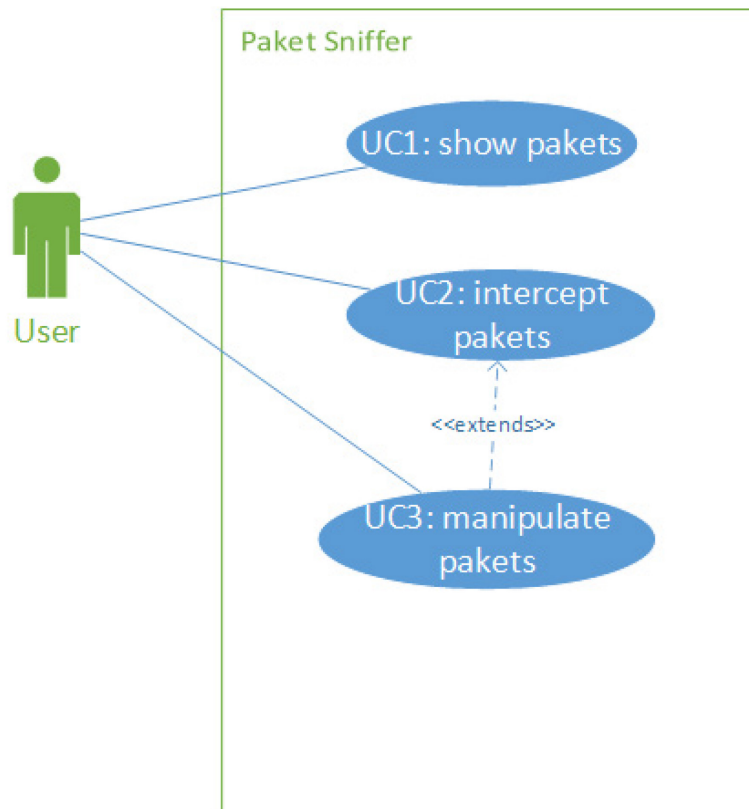


Abbildung 4: Use-Case-Diagramm

Use-Case	Beschreibung
UC1: show packets	Es werden alle Pakete im Netzwerk angezeigt
UC2: intercept packets	Es werden Pakete abgefangen
UC3: manipulate packets	Abgefangene Pakete können manipuliert und weitergesendet werden

## Sequenzdiagramme

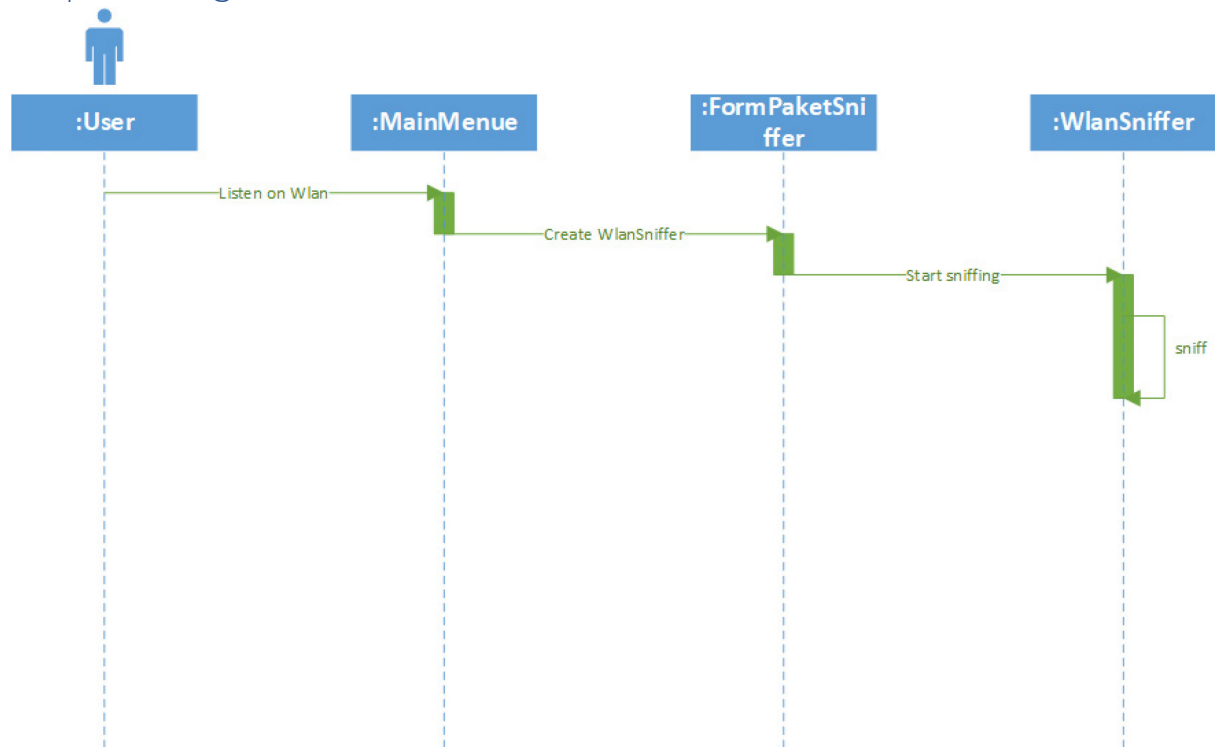


Abbildung 5: Sequenzdiagramm

Der Nutzer wählt im Hauptmenü auf welchem Interface er mithören möchte. Darauf hin wird ein Listener auf dem gewählten Interface erstellt.



## GUI-Mockups

### Main Menu – Add a listener



Abbildung 6: GUI-Mockup 1

Mit klicken auf das "+"-Zeichen erscheint ein neuer Tab. In einem Tab kann der Nutzer anwhälen, auf welchem Netzwerkinterface er mithören möchte.

## MainMenu – Listener



Abbildung 7: GUI-Mockup 2

Nachdem ein Interface ausgewählt wurde, erscheint die obige Ansicht. In der Tabelle werden alle Pakete und Informationen darüber angezeigt. Die Buttons unter der Tabelle haben verschiedene Funktionen.

Button	Ausgelöste Funktion
Inspect	Öffnet ein Fenster in dem detaillierte Informationen zum in der Liste angewählten Package angezeigt werden.
Show Stream	Es Öffnet sich ein Fenster indem die gesamte zusammenhängende Kommunikation geschrieben steht. (Ähnlich wie «Focus»)
Focus	In der Tabelle werden alle mit dem

	Package zusammenhängende Packages angezeigt. (Request und Antworten, etc...)
Edit	Erlaubt es, abgefangene Packages zu manipulieren.

## Reflexion

Das Projekt konnten wir leider nicht vollständig abschliessen. Folgendes sind Sachen, die schwer zu implementieren waren:

- Zu wenige Infos im Internet, beispielsweise zum Auslesen von Netzwerkpaketen mit Java, spezifischere kleinere Probleme mit GUI etc.
- Austausch zwischen GUI und Sniffer war schwierig hinzubekommen, wurde aber mit einem Observer-Pattern gelöst

Nächstes Mal können wir beim Projekt folgendes besser machen:

- Ein einfacheres Projekt ins Visier nehmen und in die Tiefe gehen, statt etwas schwieriges nur halb hinzukriegen
- schneller mit dem Programmieren beginnen
- bessere Zeitaufteilung

## Abbildungsverzeichnis

Abbildung 1: Klassendiagramm Netzwerksniffer	4
Abbildung 2: Grafische Darstellung des Aufbaus eines gesniffen Pakets	5
Abbildung 3: Grafische Darstellung des Aufbaus des VPN-Services	6
Abbildung 4: Use-Case-Diagramm	7
Abbildung 5: Sequenzdiagramm	8
Abbildung 6: GUI-Mockup 1	9
Abbildung 7: GUI-Mockup 2	10