Dokumentation Projekt Netzwerksniffing

Ursprüngliche Autoren: Grigory Pavlov, Colin van Loo

Datum der Erstellung: 11.09.2020

Letzter Autor: Colin van Loo

Datum der letzten Bearbeitung: 25.09.2020

# Inhaltsverzeichnis

[Inhaltsverzeichnis 1](#_Toc54343476)

[Änderungskontrolle 2](#_Toc54343477)

[Dokumentspezifikation 3](#_Toc54343478)

[Projektbeschreibung 3](#_Toc54343479)

[Funktionale Anforderungen 3](#_Toc54343480)

[Nicht-Funktionale Anforderungen 3](#_Toc54343481)

[Klassendiagramm 4](#_Toc54343482)

[Use-Cases 5](#_Toc54343483)

[Sequenzdiagramme 6](#_Toc54343484)

[GUI-Mockups 7](#_Toc54343485)

[Main Menu – Add a listener 7](#_Toc54343486)

[MainMenu – Listener 8](#_Toc54343487)

[Abbildungsverzeichnis 10](#_Toc54343488)

[Quellenangabe 10](#_Toc54343489)

# 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 | Entwurf des Pflichtenhefts | Colin van Loo | 18.09.2020 |
| 0.3 | Anpassung des Pflichtenhefts an die neue Version mit VPN. | Colin van Loo | 30.01.2020 |

# 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.

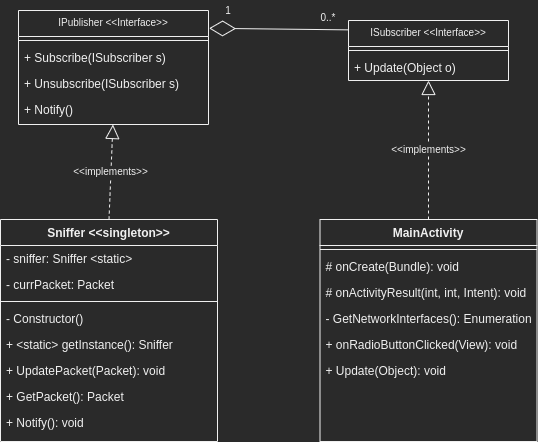
## Funktionale Anforderungen

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

## Nicht-Funktionale Anforderungen

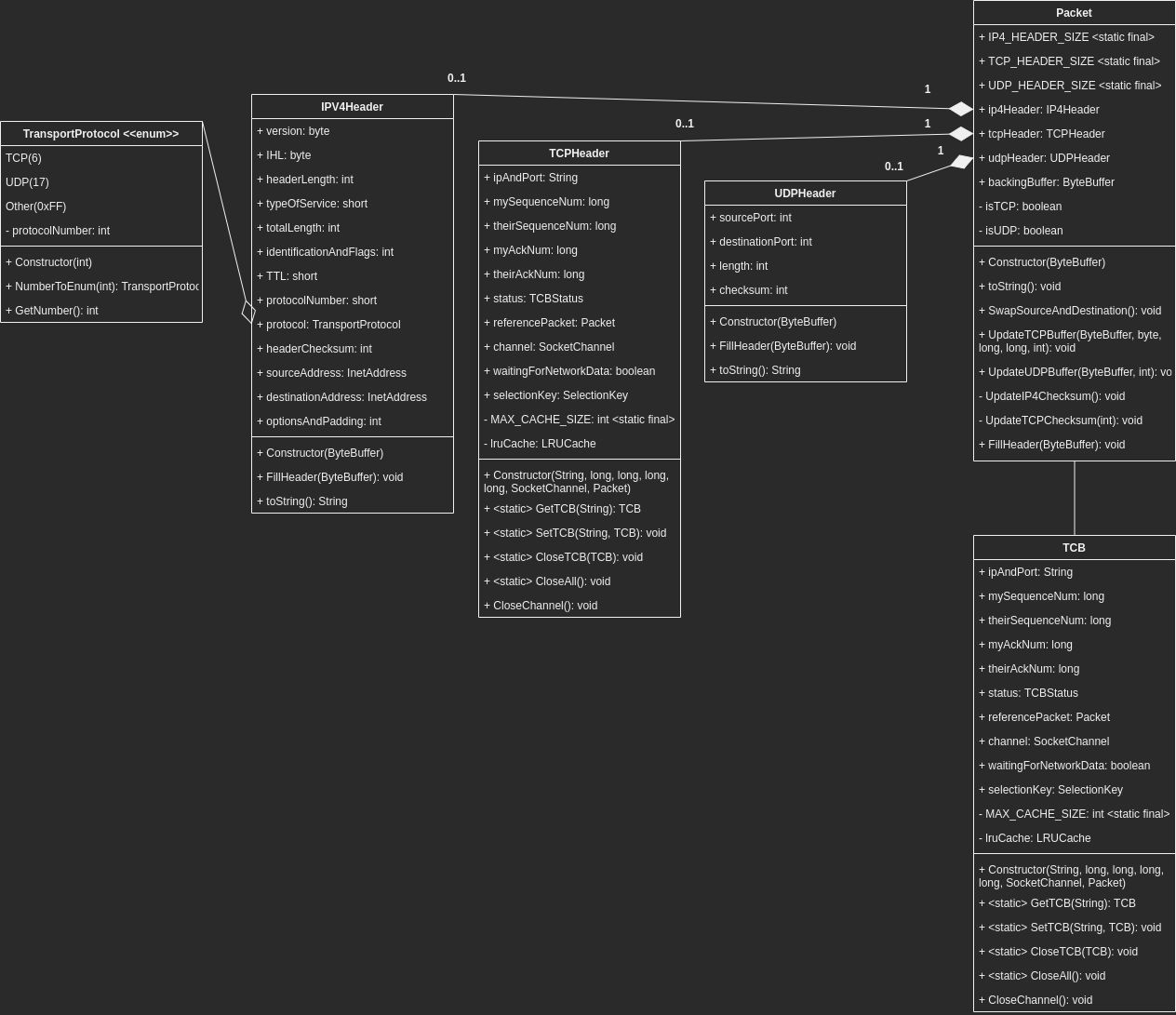
Es sollen zum Projekt UML-Diagramme erstellt werden. Das Projekt muss innerhalb dieses Modules (226a) nicht fertig gestellt werden, sondern es kann im Modul 226b weitergearbeitet werden.

## Klassendiagramm

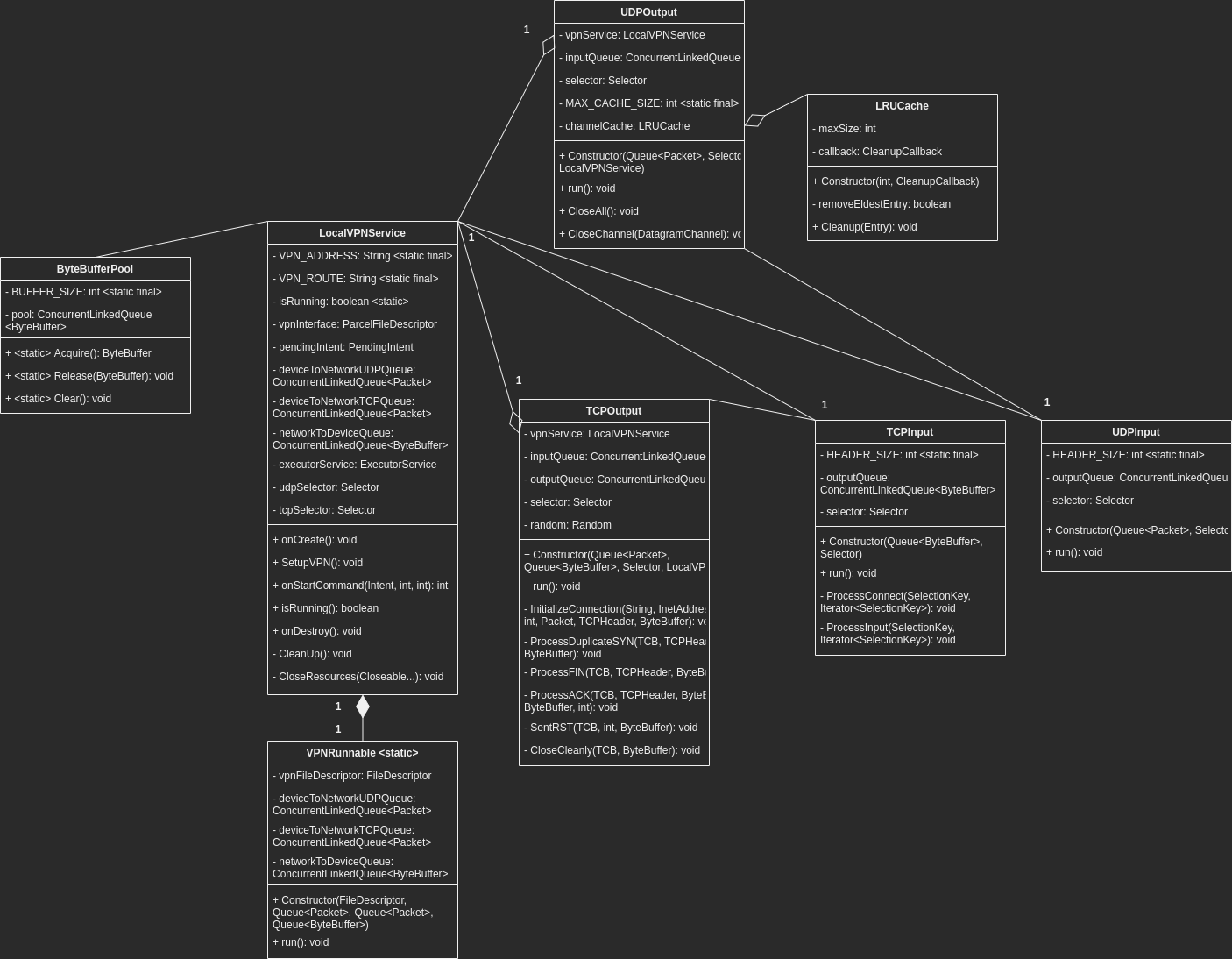


|  |  |
| --- | --- |
| **Klasse** | **Beschreibung** |
| IPublisher | Interface fuer das observer-pattern |
| Isubscriber | Interface fuer das observer-pattern |
| Sniffer | Empfaengt die packets from vpn und sendet sie an alle Subscriber. Ist eine Singleton-Klasse. |
| MainAcitvity | Main-entry-point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

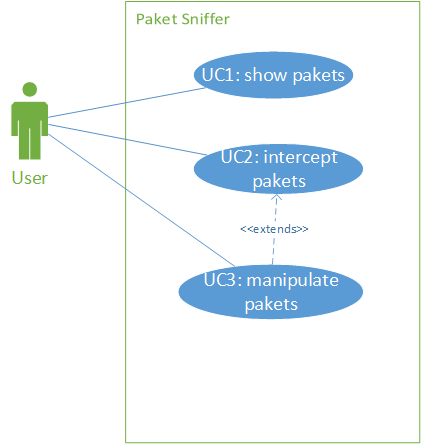
Aufbau eines Packets



Aufbau des VPN-Service

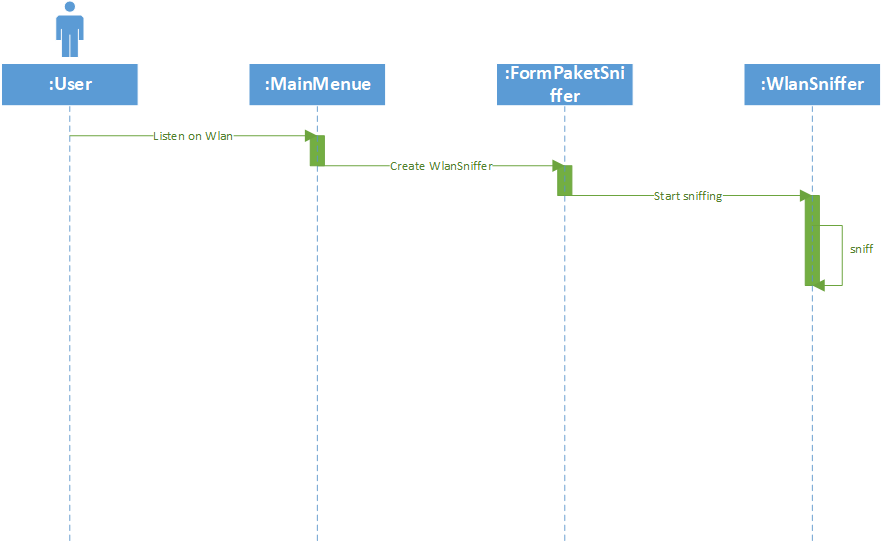


## Use-Cases



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

## Sequenzdiagramme



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



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

### MainMenu – Listener



Nachdem ein Interface ausgewählt wurde, erscheint die obige Ansicht. In der Tabelle werden alle Packete 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 angewählen 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. |

# Abbildungsverzeichnis

**No table of figures entries found.**

# Quellenangabe