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](#_Toc51918973)

[Änderungskontrolle 2](#_Toc51918974)

[Dokumentspezifikation 3](#_Toc51918975)

[Projektbeschreibung 3](#_Toc51918976)

[Funktionale Anforderungen 3](#_Toc51918977)

[Nicht-Funktionale Anforderungen 3](#_Toc51918978)

[Klassendiagramm 4](#_Toc51918979)

[Use-Cases 5](#_Toc51918980)

[Abbildungsverzeichnis 5](#_Toc51918981)

[Quellenangabe 5](#_Toc51918982)

# Änderungskontrolle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Beschreibung | Software Release | Verfasser | Datum |
| 0.1 | Entwurf des Pflichtenhefts | - | Grigory Pavlov | 11.09.2020 |
| 0.2 | Entwurf des Pflichtenhefts | - | Colin van Loo | 18.09.2020 |

# 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.

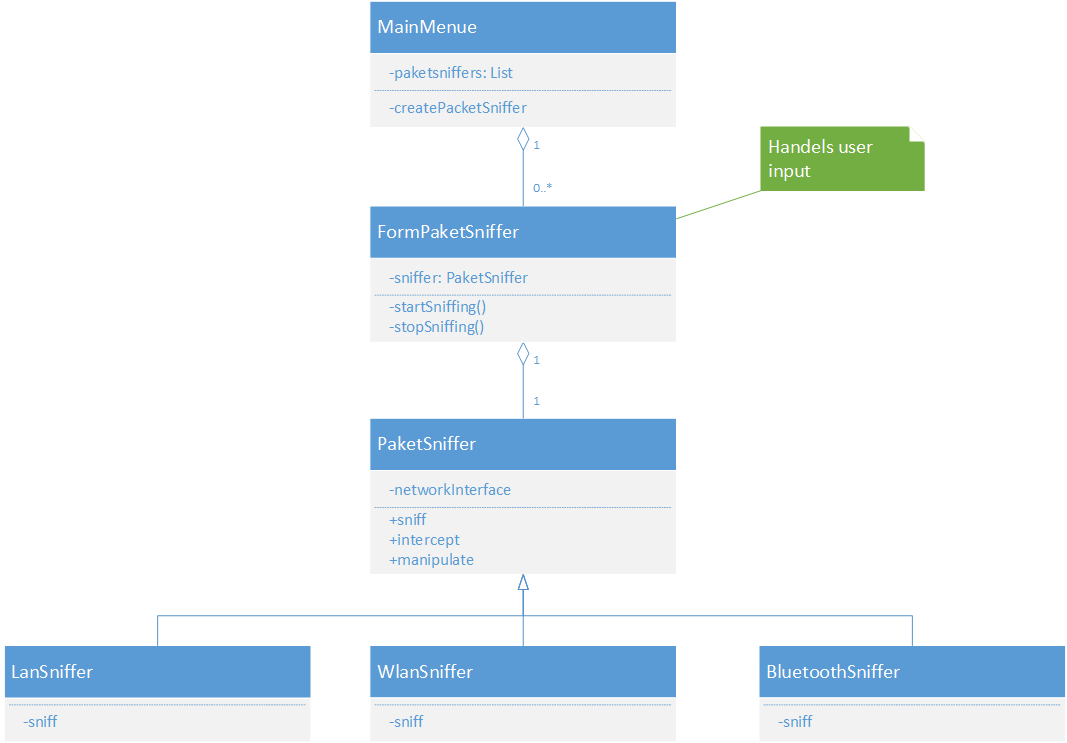
## Funktionale Anforderungen

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

## Nicht-Funktionale Anforderungen

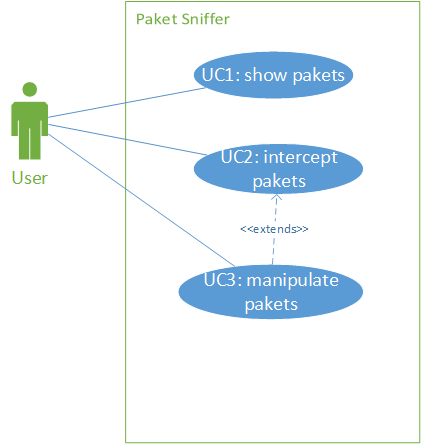
Es sollen zum Projekt UML-Diagramme erstellt werden. Das Projekt muss innerhalb dieses Modules (226a) nicht fertig gestellt werden, sondern es kann im Modul 226b weitergearbeitet werden.

## Klassendiagramm



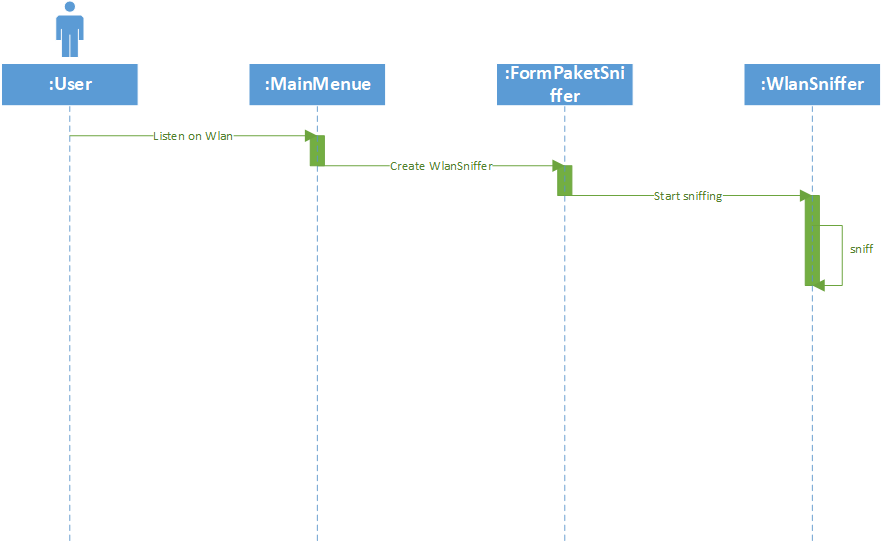
|  |  |
| --- | --- |
| Klasse | Beschreibung |
| MainMenue | Das Hauptmenü. Darin kann der Benutzer auswählen, bei welchem Netzwerk-Interface er mithören möchte. |
| FormPaketSniffer | Nachdem der Benutzer ein Interface zum mithören angewählt hat, wird ein FormPaketSniffer erstellt, der auf diesem Interface hört. Die FormPaketSniffer erhält die Nutzereingaben und bearbeitet diese. Es können mehrere FormPacketSniffer erstellt werden und somit auf verschiedenen Interfaces gleichzeitig zuhören. |
| PaketSniffer | (Parent-Klasse) Der PaketSniffer beinhaltet alle Methoden um mitzuhören, abzufangen und manipulieren. |
| LanSniffer | Beinhaltet spezifische Methoden um im Lan mitzuhören. |
| WlanSniffer | Beinhaltet spezifische Methoden um im Wlan mitzuhören. |
| BluetoothSniffer | Beinhaltet spezifische Methoden um im Bluetooth mitzuhören. |

## 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



### MainMenu – Listener



# Abbildungsverzeichnis

**No table of figures entries found.**

# Quellenangabe