

Höhere Technische Bundeslehranstalt Wien 3, Rennweg
IT & Mechatronik
HTL Rennweg: Rennweg 89b
A-1030 Wien, Tel +43 1 24215-10, Fax DW 18

Diplomarbeit

NetView

ausgeführt an der
Höheren Abteilung für
Informationstechnologie/Netzwerk- & Medientechnik
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2014/2015

durch
Martin Hammerbacher
Gregor Ivancsics
Sabine Schimpf
Abigail Steinhardt

unter der Anleitung von

Werner Lugschitz
Martin Mair
Christian Schöndorfer
Miki Zehetner

Wien, Dezember 2014

Kurzfassung

In unserem Diplomarbeitsprojekt NetView haben wir uns mit der Visualisierung und Messung von Bandbreitenauslastung im Schulnetz befasst.

Hauptkomponenten des Produkts sind eine Animation des Schulnetzes, ein Java-Programm das Bandbreitendaten im Schulnetzwerk abfragt und die Auslastungen berechnet sowie eine zentrale Datenbank als Schnittstelle zwischen diesen. Dazu wurde auch Marketing betrieben, u. a. durch ein Promotion-Video.

Die Animation stellt das Schulgebäude, die Netzwerkkomponenten (Switches, Router, Access-Points) und die Leitungen zwischen diesen dar. Je höher eine Leitung ausgelastet ist, desto stärker wird sie farblich hervorgehoben. Über eine schulinterne Website, können Benutzer die Animation sehen.

Um die three.js-Animation mit den benötigten Daten zu versorgen, gibt es einen Dienst im Netzwerk, SNMP-Manager genannt. Dieser sendet periodisch SNMP-Abfragen an die gewünschten Netzwerkgeräte. Die antworten mit bandbreitenrelevanten Werten, durch die die Bandbreitenauslastungen berechnet werden können. Die Ergebnisse werden in der zentralen MySQL-Datenbank gespeichert, wo sie der Animation zu Verfügung stehen.

Abstract

Our project NetView aims to provide a solution for bandwidth-utilization monitoring specifically for our school's network. The main components are a visual representation of the network, a bandwidth-utilization monitoring software and a central database. We also put an emphasize to marketing our product, for example through a promotion video.

The animation, written in three.js, includes a basic overview of the interior of our school, the network devices and the animated connections between those devices. The higher a connections' current utilization, the more emphasize is put on it by coloring. This is embedded on a school-internal website.

To provide the visual platform with actual data, the SNMP-Manager measures the bandwidth-utilization of specified devices and sends this gathered bandwidth-information to the database. The user of our software has to put a list of targeted routers, switches or access points, in the database, to specify which devices he/she wants to monitor.

Ehrenwörtliche Erklärung

Ich versichere,

- ❖ dass ich meinen Anteil an dieser Diplomarbeit selbstständig verfasst habe,
- ❖ dass ich keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe
- ❖ und mich auch sonst keiner unerlaubten Hilfe bzw. Hilfsmittel bedient habe.

Wien, am 23.04.2015

Präambel

Die Inhalte dieser Diplomarbeit entsprechen den Qualitätsnormen für „Ingenieurprojekte“ gemäß § 29 der Verordnung des Bundesministers für Unterricht und kulturelle Angelegenheiten über die Reife- und Diplomprüfung in den berufsbildenden höheren Schulen, BGBl. Nr. 847/1992, in der Fassung der Verordnungen BGBl. Nr. 269/1993, Nr. 467/1996 und BGBl. II Nr. 123/97.

Liste der betreuenden Lehrer

Prof. Ing. Werner Lugschitz

Prof. DI. Miki Zehetner

Prof. DI. Martin Mair

Prof. DI. Christian Schöndorfer

Liste der Kooperationspartner:

Höhere Technische Bundeslehranstalt Wien 3 Rennweg

Inhaltsverzeichnis

1	EINLEITUNG.....	8
1.1	Projektidee	8
1.2	Projektteam.....	8
1.2.1	Sabine Schimpf.....	8
1.2.2	Martin Hammerbacher	9
1.2.3	Gregor Ivancsics	9
1.2.4	Abigail Steinhardt	9
2	PROJEKTKULTUR	9
2.1	Dokumentenablage	9
2.2	Kommunikation.....	10
2.2.1	Kurze Besprechungen.....	10
2.2.2	Jour fixe	10
2.2.3	Meetings.....	10
2.2.4	Mailverkehr	11
3	PLANUNG	11
3.1	Projektziele	11
3.1.1	Muss-Ziele	11
3.1.2	Kann-Ziele	12
3.1.3	Nicht-Ziele.....	12
3.2	Projektmanagement	12
3.2.1	Zeitplanung.....	12
3.3	Meilensteine.....	14
4	VERWENDETE TECHNOLOGIEN.....	15
4.1	Three.js.....	15
4.1.1	Browser-Support	15
4.2	XAMPP	16
4.3	Adobe Illustrator	17
4.4	Adobe After Effects	17
4.5	HTML5	18
4.5.1	Neuerungen.....	18
4.5.2	Vorteile von HTML5.....	18
4.6	CSS3	19

4.6.1	Vorteile.....	19
4.6.2	Nachteile.....	19
4.7	Hosting	19
4.8	SNMP	19
4.8.1	Funktionsweise.....	20
4.8.2	Management Information Base	20
4.8.3	SNMP-Versionen	20
5	PROGRAMMIERUNG.....	22
5.1	Der SNMP-Manager.....	22
5.1.1	Einführung.....	22
5.1.2	Bandbreitenmessung und Auslastung.....	22
5.1.3	Aufbau.....	24
5.2	3D-Umgebung, Web-Interface	33
5.2.1	Szenenaufbau in Three.js.....	33
5.2.2	Kamera	33
5.2.3	Modelle.....	35
5.2.4	Antialiasing.....	39
5.2.5	Werte.....	39
5.2.6	Aufbau des dreidimensionalen Raums.....	39
5.2.7	Übersichtlichkeit.....	42
5.3	Timeline.....	43
6	DATENBANK.....	45
6.1	Anforderungen.....	45
6.1.1	Wahl der Datenbanksoftware	45
6.2	Modell.....	45
6.3	Datenstruktur.....	47
6.4	Datenbankeinträge.....	47
6.4.1	Notwendige Einträge.....	48
6.5	Timestamp	50
6.6	Schnittstelle Datenbank, SNMP-Manager	51
7	MARKETING.....	53
7.1	Logo	53
7.2	Schrift.....	54
7.3	Hintergrund.....	54
7.4	Social Media.....	54

7.4.1	Facebook.....	54
7.4.2	Twitter	55
7.5	Website	55
7.6	Promotionvideo	59
7.6.1	Allgemein.....	59
7.6.2	Inhalt	60
7.6.3	Videoerstellung.....	61
7.6.4	Szenen.....	62
7.6.5	Zusammenfügen.....	67
7.6.6	Rendering.....	67
AUTORENVERZEICHNIS		75

1 Einleitung

1.1 Projektidee

Die Idee zu unserem Projekt ist aus dem Wunsch entstanden, die beiden Fachrichtungen Netzwerktechnik und Medientechnik zu verbinden. Da unsere Klasse typengemischt ist, stellte das kein Problem dar. Je zwei Mitglieder aus jeder Fachrichtung wollten zeigen, was sie in den vergangenen Jahren gelernt haben und was die jeweilige Fachrichtung zu bieten hat.

Die erste Idee für das Diplomarbeitsthema war, die Animation eines Themas aus der Netzwerktechnik zu erstellen. Nach langer Überlegung einigte sich das Team auf eine Echtzeitdarstellung des Schulnetzes, so wie es im Schulgebäude aufgebaut ist. Dabei soll die Auslastung einer Leitung zwischen zwei Geräten farblich dargestellt werden und die Daten sollen in möglichst kurzen Abständen aktualisiert werden.

Die fertige Anwendung soll ein Hilfsmittel zur Optimierung des Netzwerks hinsichtlich auftretender Probleme und Fehler sein.

1.2 Projektteam

1.2.1 Sabine Schimpf

Ihr Aufgabenbereich bei der Umsetzung bestand in der Erstellung der 3D-Umgebung und der Programmierung des Web-Interfaces. Aufgrund ihrer persönlichen Erfahrungen während und außerhalb des Unterrichts und ihrer Begeisterung an der 3D-Animation stellte dieser Aufgabenbereich kein Problem hinsichtlich der Motivation dar. Dadurch, dass die verwendete Software auch im Unterrichtsfach MPCA eingesetzt wurde und andere Diplomarbeiten ähnliche Programme benutzten, standen bei Problemen immer ausreichend Ansprechpartner zur Verfügung.

Als Projektleiterin der Diplomarbeit gehörten auch das Projektmanagement und die Motivation des Teams zu ihren Aufgabenbereichen. Da jedes Teammitglied aber bis Ende des Projektes in seinem jeweiligen Aufgabenbereich ausreichend motiviert war, gab es in diesem Bereich kaum Schwierigkeiten.

1.2.2 Martin Hammerbacher

Im Zuge der Diplomarbeit war ich vor allem für die Konzipierung der Datenbank, sowie für die Weiterverarbeitung und Strukturierung der zuvor eingelesenen Daten verantwortlich. Hierbei konnte ich meine Erfahrungen aus den Fächern Datenbanksysteme und Netzwerkprogrammierung, sowie meiner Ferialjobs zurückgreifen.

Außerdem hatte ich die Rolle des stellvertretenden Projektleiters inne. Am meisten Freude hat mir die Zusammenarbeit zwischen unseren Fachrichtungen bereitet, wobei ich viele Erfahrungen aus beiden Bereichen sammeln konnte.

1.2.3 Gregor Ivancsics

Meine Hauptaufgabe bestand in der Programmierung des netzwerktechnischen Teils der Arbeit. Ziel meiner Tätigkeit war es, ein Java-Programm zu schreiben, das die 3D-Animation mit Daten zur Visualisierung des Schulnetzes versorgt. Hierbei konnte ich eine Vielzahl an gelernten Fähigkeiten aus fünf Schuljahren einsetzen. Die Fächer Allgemeine- bzw. Netzwerkprogrammierung und Netzwerkmanagement spielten dabei die größte Rolle.

Die Erfahrungen in der Softwareentwicklung, bereiteten mir die meiste Freude, wodurch ich in diesem Bereich einiges dazulernte.

1.2.4 Abigail Steinhardt

Meine Aufgabe in der Diplomarbeit war das Marketing. Im Zuge dessen habe ich eine Website erstellt und auf dem neusten Stand gehalten, sowie Social-Networking Seiten erstellt und diese betreut. Meine Hauptaufgabe war es, ein Promotion Video zu erstellen, welches dem Nutzer hilft zu verstehen um was es bei NetView geht und wozu man es brauchen kann.

2 Projektkultur

2.1 Dokumentenablage

Zur Verwaltung der Dokumente wurde Dropbox verwendet. Dropbox ist eine Gratissoftware, die es ermöglicht von überall auf unsere Dokumente zuzugreifen. Zusätzlich sind die Dokumente vor Verlust geschützt, da auch gelöschte Daten mit wenigen Klicks wiederhergestellt werden können. Dropbox wurde bereits im Unterricht zur schnellen und einfachen Verteilung von Informationen und zum Verwalten von Daten verwendet.

Um die Übersichtlichkeit zu garantieren, wurde eine einfache Ordnerstruktur erstellt. Zusätzlich wurde jedes erstellte Dokument wie folgt benannt:

NV_Dokument_[Datum/Version]

(Diplomarbeits-Kürzel, Name des Dokumentes und optional Datum oder Versionsnummer)

Zur Verwaltung und Versionierung der Software wurde Git eingesetzt. Git wurde im Projektmanagement-Unterricht vorgestellt und erwies sich auch in anderen Projekten als praktisch. Durch die stark getrennten Bereiche in der Programmierung wurde der Code von jedem Teammitglied selbst verwaltet.

2.2 Kommunikation

Die Kommunikation ist entscheidend für das Scheitern oder den Erfolg eines Projektes. Um eine reibungslose Kommunikation im Team, mit den Betreuern und mit externen Personen zu gewährleisten wurden folgende Regeln festgelegt:

2.2.1 Kurze Besprechungen

Die Kommunikation im Projektteam erfolgt hauptsächlich verbal und via Kurznachrichten über WhatsApp. Bei kurzen Besprechungen werden die Ergebnisse als kurze Nachricht über WhatsApp festgehalten.

2.2.2 Jour fixe

Die Teilnehmer eines Jour fixes sind der Projektleiter und alle Mitglieder des Projektteams, optional ist auch der Hauptbetreuer anwesend. Besprochen werden unter anderem der Stand des Projektes, der weitere Verlauf und die nächsten Ziele. Sollten Probleme, ob technischer oder sozialer Natur, aufgetreten sein sind auch diese zu behandeln und zu lösen. Am Ende eines Jour fixes wird eine kurze Notiz mit den Ergebnissen an alle Teilnehmer gesendet.

2.2.3 Meetings

Ein Meeting kann jederzeit vom Auftraggeber (Hauptbetreuer), Projektleiter als auch von jedem Teammitglied einberufen werden. Während eines Meetings berichtet jedes Teammitglied über den Fortschritt und aufgetretene Probleme. Am Ende eines jeden Meetings werden die nächsten Schritte definiert, Termine gesetzt und optional auch das nächste Meeting geplant. Nach Abschluss des Meetings wird ein Protokoll an alle Teilnehmer gesendet.

2.2.4 Mailverkehr

Beim Verfassen von E-Mails sollte generell auf eine einheitliche Form geachtet werden. Grundsätzlich werden E-Mails immer an das ganze Projektteam gesendet. Nachrichten an externe Personen werden vor dem Absenden vom Hauptbetreuer Korrektur gelesen.

3 Planung

3.1 Projektziele

3.1.1 Muss-Ziele

- Die Teammitglieder eignen sich das für die Umsetzung der Diplomarbeit notwendige Wissen an.
- Dem Projektteam steht ein passender Server für das Web-Interface, die Datenbank sowie zum Auslesen der Daten zur Verfügung.
- Zur Verwaltung der Daten ist eine Datenbank in einem zuvor gewählten Datenbanksystem erstellt.
- Daten über die Auslastung der Switches oder Access Points der Räume werden periodisch erfasst.
- Die ausgelesenen Daten werden in die Datenbank eingetragen
- Ein 3D-Modell des Schulnetzes ist erstellt.
- Ein Konzept zur Auswertung der gesammelten Daten ist entwickelt.
- Über ein Web-Interface werden regelmäßig Daten aus der Datenbank ausgelesen und dreidimensional dargestellt.
- Eine Social-Network-Seite ist erstellt und wird im Laufe des Projekts auf dem neusten Stand gehalten.
- Ein Promotion Video ist erstellt.

3.1.2 Kann-Ziele

- Die Ausbildung zum Cisco Certified Network Associate ist von den Teammitgliedern der Fachrichtung „Netzwerktechnik“ erfolgreich abgeschlossen.
- Das Web-Interface wird plattformunabhängig korrekt dargestellt (Responsive Design).
- Sponsoren für alle eventuell anfallenden Kosten sind gefunden.
- Durch das Interface können Daten von jedem beliebigen Zeitpunkt eingesehen werden.

3.1.3 Nicht-Ziele

- Typ und Inhalt der verkehrenden Daten werden ausgewertet.
- Durch das Projekt wird kommerzieller Ertrag erzielt.
- Neue Netzwerkgeräte werden vom Projektteam angeschafft.

3.2 Projektmanagement

Durch die ungewöhnliche Zusammensetzung unseres Teams und die getrennten Arbeitsbereiche war es jedem Teammitglied möglich, sich komplett auf seine eigene Arbeit zu konzentrieren, ohne vom Fortschritt der anderen betroffen zu sein. Für das Projektmanagement in unserer Diplomarbeit haben wir daher eine eigene Methode, basierend auf Scrum, entwickelt, bei der jedes Mitglied selbst für seine Planung und sein Zeitmanagement verantwortlich ist.

Zu Beginn des Projektes wurde von jedem Teammitglied eine Liste mit seinen jeweiligen Tasks erstellt. Diese Liste wurde im Laufe des Projektes individuell erweitert und angepasst, zum Beispiel bei neuen Erkenntnissen oder nach einer Besprechung.

3.2.1 Zeitplanung

Unser Projekt war in mehrere Phasen aufgeteilt. Anders als die Sprints bei Scrum hatten diese Phasen aber keinen fix definierten Zeitraum. Beginn und Ende einer Phase wurden durch ein Meeting mit dem Hauptbetreuer festgelegt. Bei jedem Meeting wurden der Fortschritt und die nächsten Schritte im Projekt besprochen. Aus den Ergebnissen stellte sich jedes Teammitglied seine Tasks für die nächste Phase aus seiner Taskliste zusammen und ergänzte diese nach Bedarf. Die Dauer wurde für jede Umsetzungsphase individuell durch den Termin für das nächste Meeting definiert.

So entstanden folgende Phasen:

- *Planungsphase (Mai 2014 - September 2014):*
 - Sammeln von Ideen, Erstellung des Ansuchens
 - Erstellen von Planungsdokumenten
- *Umsetzungsphase 1 (September 2014 - Oktober 2014):*
 - Sammeln von Erfahrung mit der Software
 - Erste Schritte bei der Programmierung
- *Umsetzungsphase 2 (Oktober 2014 - Dezember 2014):*
 - Programmierung der Software
 - Erste Tests
 - Lösen von diversen Problemen
- *Umsetzungsphase 3 (Jänner 2015 - März 2015):*
 - Fertigstellung der Programmierung
 - Umsetzung der ersten Kann-Ziele
 - Erstellen des Diplomarbeitsbuches
- *Umsetzungsphase 4 (März 2015 - April 2015):*
 - Fertigstellung des Diplomarbeitsbuches
 - Feinschliff an der Software
 - Vorbereitung auf die Matura
- *Abschlussphase (April 2015 - Juni 2015):*
 - Veröffentlichung der Software

- Abgabe des Diplomarbeitbuches
- Matura

Trotz anfänglicher Schwierigkeiten in den ersten beiden Umsetzungsphasen traten im weiteren Verlauf des Projektes keine weiteren zeitlichen Probleme auf.

3.3 Meilensteine

Datum	Meilenstein
19.09.2014	Social-Network-Seite erstellt
19.09.2014	Webpräsenz erstellt
17.10.2014	Erste Version der Oberfläche fertiggestellt
17.10.2014	Erster Prototyp fertiggestellt
14.11.2014	Promotion-Video veröffentlicht
14.11.2014	Oberfläche überarbeitet
14.11.2014	Zweiter Prototyp fertiggestellt
12.12.2014	Präsentation vorbereitet
12.12.2014	Dritter Prototyp fertiggestellt
27.02.2014	Produkt fertiggestellt
31.05.2015	Abschluss des Projektes

Tabelle 1: Im Projektantrag definierte Meilensteine

Das Team setzte sich zu Beginn des Projektes beim Erstellen des Antrages Meilensteine für die Umsetzung. Durch die Verzögerungen bei der Umsetzung und die Änderungen in der Art der Umsetzung wurden die Meilensteine teilweise zu geänderten Zeitpunkten erfüllt.

4 Verwendete Technologien

4.1 Three.js

Three.js ist eine kostenlose JavaScript-Bibliothek unter der MIT-Lizenz [MIT] zum Erstellen und Animieren von 3D-Grafiken in Browsern. Für die Verwendung von Three.js werden keine weiteren Programme oder Browser Add-Ons benötigt. Three.js basiert auf WebGL und läuft daher in allen Browsern, die WebGL unterstützen.

4.1.1 Browser-Support

Liste aller Browser in der aktuell meist verwendeten Version mit WebGL-Unterstützung [CIU]:

Desktop-Browser

- Internet Explorer, Version 11
- Firefox (Partieller Support), Version 35
- Google Chrome, Version 40
- Safari (Partieller Support), Version 8
- Opera (Partieller Support), Version 27

Mobile Browser

- iOS Safari, Version 8.1
- Android Browser (Partieller Support), Version 37
- Chrome for Android (Partieller Support), Version 40

Partielle Unterstützung bedeutet, dass nicht alle Benutzer dieses Browsers WebGL aktiviert haben. Der Grund dafür sind oft alte Treiber für ihre Grafikkarte. [BAW] Für eine genaue Liste mit allen unterstützten Browsern siehe www.canIuse.com/WebGL.

Ursprünglich wurde Three.js für ActionScript entwickelt, es wurde aber bereits kurze Zeit später in JavaScript umgeschrieben, um eine Plattformunabhängigkeit zu erreichen. Die erste offizielle Version von Three.js wurde 2010 vom Entwickler, Ricardo Cabello, auf GitHub veröffentlicht. [TJS]

4.2 XAMPP

XAMPP fungiert als lokaler Webserver, da manche Browser das Ausführen von JavaScript vom lokalen Speicher teilweise nicht zulassen. Wird der Code aber von einem Webserver abgerufen, kann dieses Verhalten umgangen werden. XAMPP ist bestens dafür geeignet, da es alle notwendigen Komponenten beinhaltet und keine zusätzliche Software installiert werden muss.

Da XAMPP bereits mehrmals zuvor im Unterricht und privat zum Einsatz gekommen war, konnten auch Probleme, unter anderem das Fehlen von notwendigen Berechtigungen zur Ausführung des Programms, schnell behoben werden. Die Anwendung ist ein Softwarepaket bestehend aus einem Apache Web Server, MySQL, PHP, Perl und einigen Verwaltungstools, wie zum Beispiel phpMyAdmin für den einfachen Umgang mit der Testdatenbank ohne SQL-Befehle, sowie ein Tool zum Aufspüren von blockierten Ports.

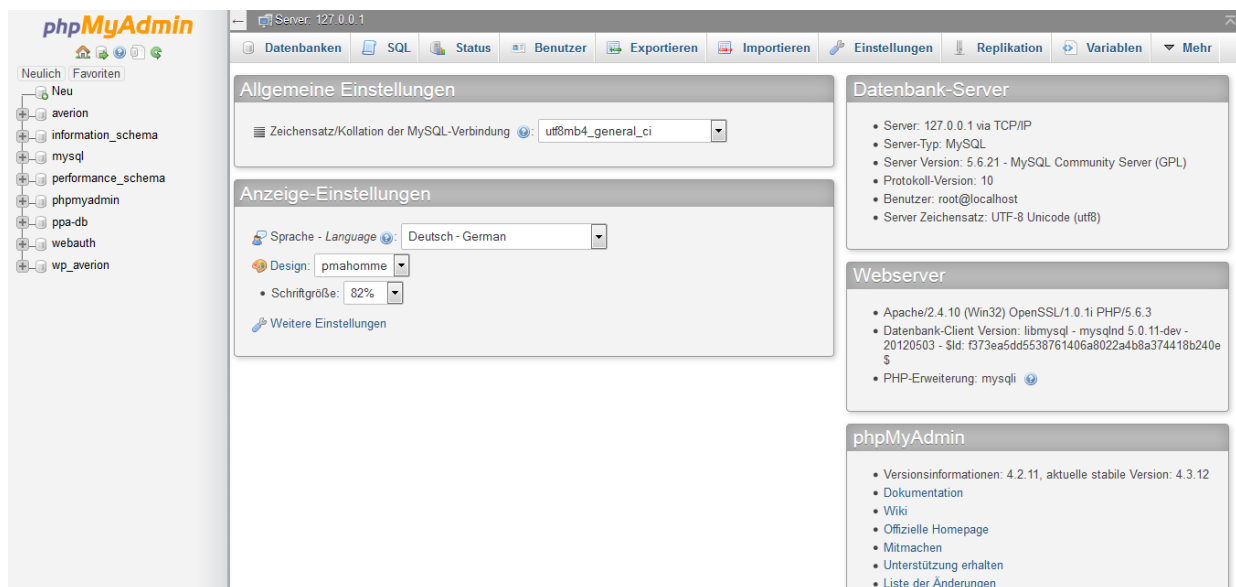


Abbildung 1: phpMyAdmin

Weitere Programme, wie zum Beispiel ein FileZilla FTP-Server, können ganz einfach als Erweiterung installiert werden.

Alle Komponenten können zentral von einem Control Panel aus gestartet und verwaltet werden. Verwendet wurde Version 5.6.3.

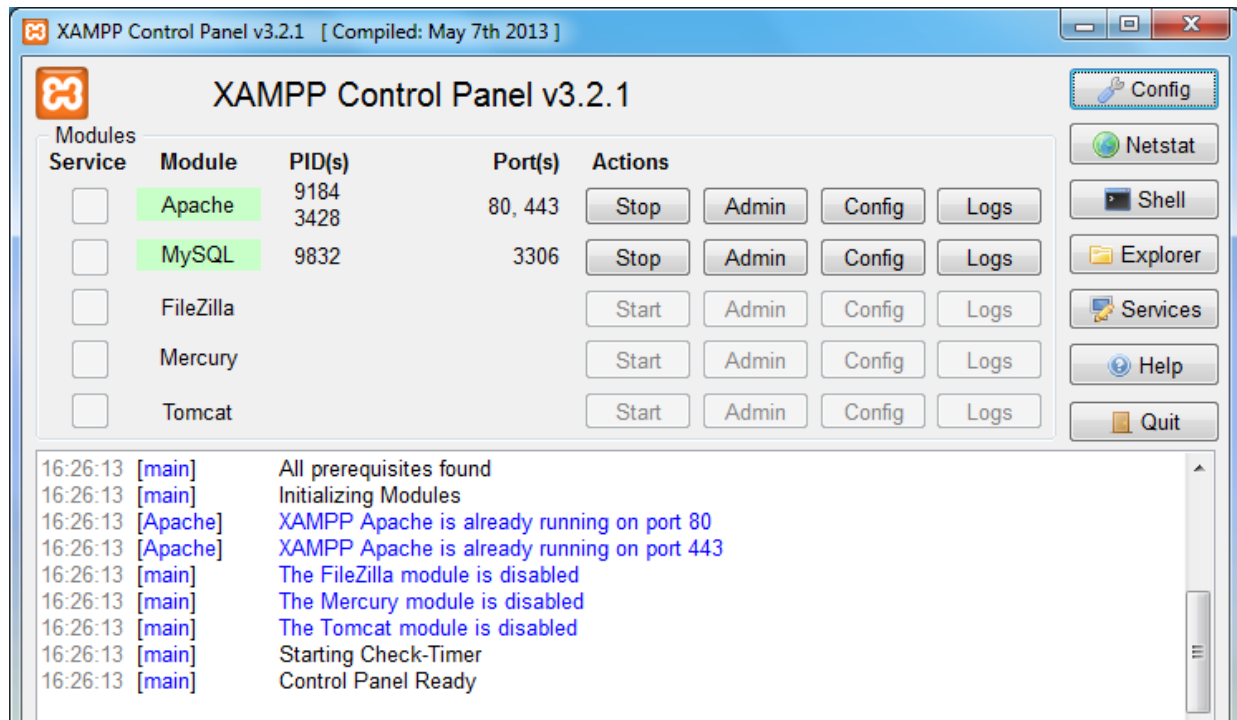


Abbildung 2: XAMPP Control Panel

4.3 Adobe Illustrator

Das NetView-Logo wurde in Adobe Illustrator umgesetzt. Adobe Illustrator (kurz Ai) ist ein vektorbasiertes Grafik- und Zeichenprogramm. Es dient also dem Herstellen von Computergrafiken, die man ohne Qualitätsverlust beliebig in ihrer Größe verändern kann – im Unterschied zu pixelbasierten Bildbearbeitungsprogrammen wie z. B. Photoshop. [AIL]

4.4 Adobe After Effects

Das Video wurde mit Adobe After Effects aus der Creative Suite CS6 erstellt. After Effects ist eine Animationssoftware die sich für Nachbearbeitung von Aufgenommenen Szenen oder Animation selbst eignet.

„Optimal geeignet ist Adobe After Effects für die Erstellung kurzer Videoclips, die mit vielen Effekten ausgestattet werden sollen, wie beispielsweise Trailer oder Werbeclips. In diese können im 2D- oder 3D-Modus nachträglich Animationen und Effekte eingefügt werden. Dabei stehen in Adobe After Effects dem Nutzer verschiedene Kameraperspektiven zur Verfügung, wodurch die Videoclips optimal mit Effekten versehen werden können. Ein weiteres Feature der Software ist die Möglichkeit, Objekte aus anderen Adobe Creative Suite-Programmen zu importieren; so können z.B. mit Photoshop bearbeitete 3D-Bilder in Videoclips eingefügt werden. [AEF]

4.5 HTML5

HTML5 ist wie HTML4 und XHTML eine Auszeichnungssprache welche auch semantisch strukturieren kann. Dank HTML5 können zum Beispiel Flash Animationen abgelöst werden.

4.5.1 Neuerungen

Die Neuerungen gegenüber der vorhergehenden Version HTML4 ist unter anderem die Ergänzung von neuen Funktionen und Elementen. Einige Beispiele für neue Elemente sind:

- `<section>`
 - Definiert den Abschnitt eines Dokuments.
- `<nav>`
 - Ein Abschnitt der nur Navigationslinks enthält
- `<article>`
 - Unabhängiger Inhalt
- `<aside>`
 - Definiert eine Randbemerkung
- `<header>`
 - Der Kopfteile eines Dokuments
- `<footer>`
 - Der Fußteil eines Elements
- `<main>`
 - Nur einmal pro Seite erlaubt, dieser tag definiert den Hauptinhalt der Seite

4.5.2 Vorteile von HTML5

Die neue Version der Auszeichnungssprache hilft Webseiten vielseitiger und auch leistungsfähiger zu machen.

Unter anderem bietet HTML5 einfacheres einbinden von Video und Ton, Optimierung von Grafiken (2D und 3D), neue Möglichkeiten zur Serverkommunikation, lokales Speichern von Userdaten (so wird auch das offline arbeiten ermöglicht), kürzere Ladezeiten und außerdem semantische Strukturierung von Dokumenten.

4.6 CSS3

CSS3 ist die nächst höhere und momentan aktuellste Version nach CSS. CSS steht für „Cascading Style Sheet“, es ist dafür da, moderne Seiten zu entwickeln und gestalten.

4.6.1 Vorteile

CSS3 bietet viele neue Möglichkeiten und Funktionen gegenüber CSS. Während der Seitenaufbau schneller funktioniert ist der Speicheraufwand geringer. Außerdem werden barrierefreie Seiten einfacher umzusetzen.

4.6.2 Nachteile

CSS3 ist, obwohl die meisten Browser es unterstützen, in vielen Browsern entweder unterschiedlich, oder nicht komplett eingebettet.

So kann es sein das eine Website in einem Browser anders aussieht als im anderen, beziehungsweise sie nicht komplett funktioniert.

4.7 Hosting

Website und Webinterface werden von einem Webserver im Schulnetz aus gehostet. Dies erspart uns sowohl die Kosten für Webspace als auch die Kosten und Einrichtung der Domain. Die Einrichtung des Webspaces hat freundlicherweise Prof. Mag. Andreas Fink für uns übernommen.

Wir erreichen unseren Webspace über FTP. Bereits während der Entwicklung konnten die ersten Ergebnisse darauf getestet und hergezeigt werden. Über den Browser sind wir unter www4.htl.rennweg.at/NetView erreichbar, Großbuchstaben müssen beim Eintippen beachtet werden.

4.8 SNMP

SNMP steht für Simple Network Management Protocol, es wird seit 1988 von der IETF (Internet Engineering Task Force) entwickelt. SNMP wurde entwickelt um Netzwerkgeräte, wie Router, Switches, Drucker, usw. von einer zentralen Station aus zu steuern und zu überwachen. SNMP regelt dabei die Kommunikation zwischen den zu überwachenden Geräten und der zentralen Überwachungsstation.

Das Protokoll wurde dafür ausgelegt jedes netzwerkfähige Gerät in die Überwachung aufzunehmen, dabei beschreibt es den Aufbau der Datenpakete, sowie den Kommunikationsablauf. Durch seine Einfachheit und vielseitige Anwendbarkeit, hat sich SNMP im Laufe der Jahre zu einem Standard entwickelt, der von nahezu allen Managementprogrammen und Endgeräten unterstützt wird.

4.8.1 Funktionsweise

Zur Überwachung der Netzwerkgeräte werden sogenannte Agents verwendet. Agents sind Programme, die direkt auf den zu überwachenden Geräten laufen, sie sind in der Lage den Zustand eines Gerätes zu beurteilen und Aktionen zu setzen. SNMP ermöglicht die Kommunikation zwischen diesen Agents und der zentralen Managementstation.

4.8.2 Management Information Base

SNMP selbst legt nicht fest welche Werte ein Gerät liefert, da sich diese unterscheiden können. Hier kommt die sogenannte Management Information Base (MIB) ins Spiel, sie beschreibt welche Werte eine Netzwerkkomponente liefern kann und ordnet diese tabellarisch an.

4.8.3 SNMP-Versionen

Da SNMP bereits seit 1988 existiert, fanden im Laufe der Jahre mehrere Weiterentwicklungen und Verbesserungen statt. In unserem Projekt wird unter folgenden Versionen unterschieden:

- SNMPv2c
- SNMPv3
 - noAuthPriv
 - AuthPriv

4.8.3.1 SNMPv2c

Nahezu alle SNMP fähigen Netzwerkgeräte unterstützen SNMPv2c, gegenüber v1 bietet v2c mehrere Vorteile, so können zum Beispiel unterschiedliche Errors erkannt und gemeldet werden. Ein weitere Neuerung im Vergleich zu Vorherigen Version ist die Einführung von sogenannten „Inform-Messages“, diese fordern für jede gesendete Nachricht eine Bestätigung.

SNMPv2c Security Model:

- Security Level: noAuthNoPriv
- Authentifizierung: Community String
- Verschlüsselung: Keine

4.8.3.2 SNMPv3

Version 3 ist der aktuellste SNMP-Standard. Im Gegensatz zu den als sehr unsicher geltenden vorherigen Version wurde bei v3 hauptsächlich auf verbesserte Security geachtet. So bietet v3 nun die zusätzlichen Features Authentifizierung und Verschlüsselung an, welche zusammen oder separat genutzt werden können. NetView unterstützt alle in der folgenden Tabelle angeführten Security Level:

Security Level	Authentifizierung	Verschlüsselung
noAuthNoPriv	Username	Keine
authNoPriv	MD5, SHA	Keine
authPriv	MD5, SHA	DES

Tabelle 2: SNMPv3-Security Level

5 Programmierung

5.1 Der SNMP-Manager

5.1.1 Einführung

Die Aufgabe, Bandbreitendaten von Switches und Routern im Schulnetzwerk zu beschaffen, lösen wir mit dem weit verbreiteten Netzwerkmanagementprotokoll „Simple Network Management Protocol“ . [SNM]

Ist SNMP auf einem Computer, Switch oder Router installiert, kann damit der Austausch von Information über das Gerät (z. B. der Hostname) geregelt werden.

Es gibt zwei Rollen die eine SNMP-Implementation annehmen kann. Die des SNMP-Managers, der die restlichen Geräte in einer SNMP-Domain ansteuern kann, ist vergleichbar mit der eines Administrators. Die anderen Hosts nehmen die Rolle eines SNMP-Agents an und lassen sich dadurch administrieren bzw. abfragen.

Unsere Software nimmt die Rolle des Managers ein. Die zu messenden Geräte im Schulnetzwerk verwenden die integrierten Agents um Bandbreitenbezogene Daten zur Verfügung zu stellen.

5.1.2 Bandbreitenmessung und Auslastung

Das Wort Bandbreite kommt aus der Nachrichtentechnik bzw. der Physik und beschreibt die Breite eines Intervalls innerhalb eines bestimmten Frequenzspektrums. [DUD]

In digitaler Kommunikation ist damit die Anzahl der Bits, die pro Sekunde über eine Netzwerkkarte traversieren, gemeint. Also die Menge an Daten, die von einer Netzwerkschnittstelle umgesetzt werden.

Anhand dieser Rate können Netzwerkadministratoren u. a. feststellen, wo welches Netzwerkequipment gebraucht wird bzw. wie „schnell“ die Internetanbindung sein sollte. Außerdem kann damit überprüft werden ob der Internet-Provider das *Service-Level* einhält.

Eine weitere interessante Metrik ist die Bandbreitenauslastung. Diese setzt die Bandbreite eines Interfaces mit seiner maximal unterstützten Bandbreite in Relation.

Unterstützt eine Netzwerkkarte maximal 10 MBit/s und verschickt und empfängt genau 5 MBit in einer Sekunde, so ist sie in dieser einen Sekunde zu 50% ausgelastet.

Es ist jedoch ausschlaggebend welche Art von Netzwerkinterface bemessen wird. Handelt es sich um ein Interface in einem Shared-Medium, also ein LAN ohne Switches in dem alle Endknoten an der selben Verbindung hängen, so ist es nicht möglich gleichzeitig zu senden und zu empfangen. Solche Schnittstellen werden als half-duplex bezeichnet.

Bei geschwitchten Ethernet-Netzwerken wie unserem Schulnetz, können Endknoten zum selben Zeitpunkt senden und empfangen. Daraus ergibt sich der Umstand, das an solchen Geräten zweierlei Bandbreitenauslastungen gemessen werden können: Eingehende und ausgehende (engl. *incoming* und *outgoing*).

Um einen dieser beiden Werte zu berechnen, muss die Anzahl der in die jeweilige Richtung fließenden Bits gemessen werden. SNMP bietet hierfür Zähler, *ifInOctets* und *ifOutOctets*, die für jedes weitergeleitete Byte (octet) um eins erhöht werden. [CI1] [CI2]

Wird einer dieser *counter* abgefragt, muss, um die Anzahl der jeweiligen Bits zu bekommen, dasselbe an einem späteren Zeitpunkt wiederholt werden. Wurde zusätzlich die in zwischen vergangen Zeit gemessen, kann die Bandbreite berechnet werden.

$$\text{Bandbreite} = \frac{\Delta \text{Bit}}{\Delta t}$$

ΔBit ist die Anzahl der im Zeitraum Δt verschickten oder empfangen Bits.

Die maximal unterstützte Bandbreite repräsentiert, außer bei WAN-Interfaces, laut Cisco der Wert *ifSpeed*. Da dieser sich während dem Betrieb nicht ändern sollte, reicht es diesen einmalig pro Interface abzurufen. [CI3]

Mit diesen drei SNMP-Werten und dem gemessenen Zeitdelta können folgende Formeln verwendet werden:

$$\text{Eingehende Bandbreitenauslastung} = \frac{\Delta \text{ifInOctets} \cdot 8 \cdot 100}{\Delta t \cdot \text{ifSpeed}}$$

$$\text{Ausgehende Bandbreitenauslastung} = \frac{\Delta \text{ifOutOctets} \cdot 8 \cdot 100}{\Delta t \cdot \text{ifSpeed}}$$

5.1.3 Aufbau

Der SNMP-Manager wurde in Java mit Verwendung des APIs „SNMP4J“ (snmp4j.org) programmiert. Für einen Klassen-Überblick ist „Abbildung 3“ da. Folgend werden die Klassen und Funktionen des Programms beschrieben.

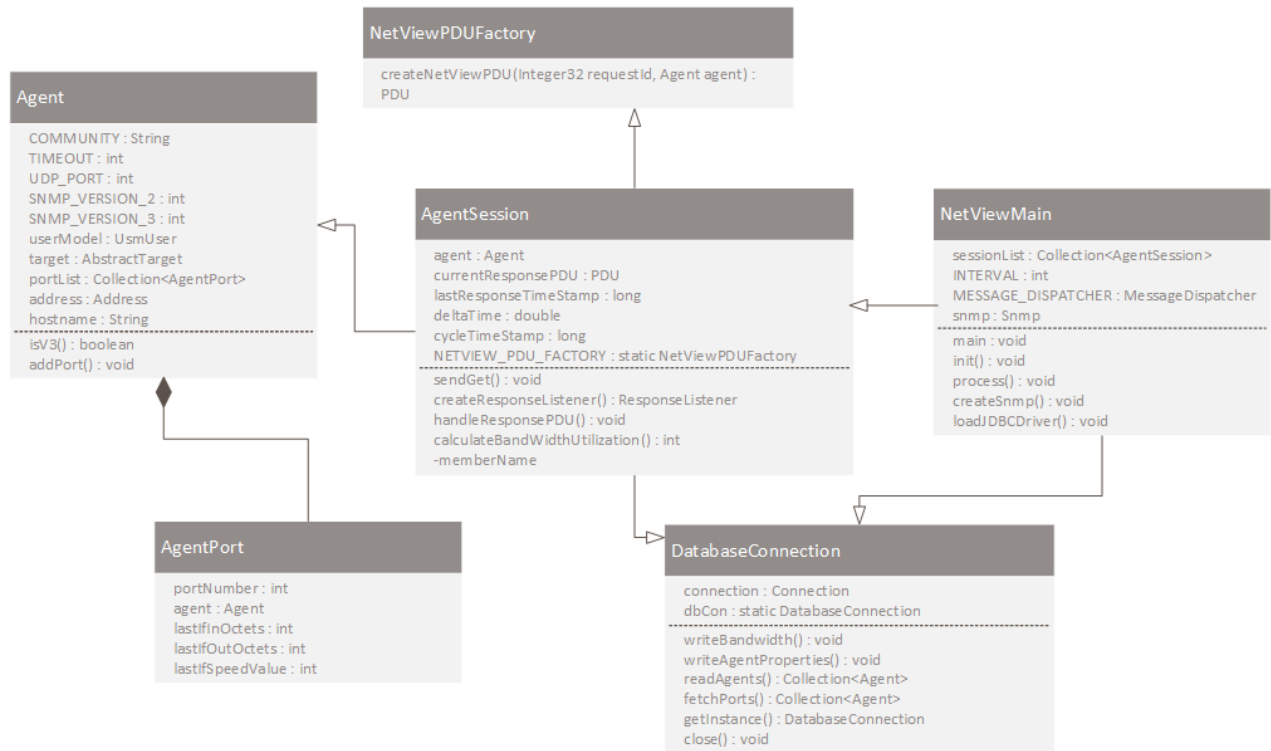


Abbildung 3: UML Klassendiagramm

5.1.3.1 NetViewMain

5.1.3.1.1 Allgemeines

Die Aufgaben der Hauptklasse NetViewMain sind das Starten der Verbindungen zwischen Manager und Agents (AgentSessions), die Steuerung der periodischen Abläufe und das Laden des JDBC-Treibers.

5.1.3.1.2 Init

Folgende *init*-Methode beschreibt das Laden des JDBC-Treibers und den Start einer Datenbankverbindung um die Liste der Agents herauszulesen. Außerdem werden hier schon die Sicherheitsdetails, in Form der UserModels, für SNMPv3 dem Snmp-Objekt (*snmp*) geliefert.

```

loadJDBCDBDriver();
DatabaseConnection dbcon = new DatabaseConnection();
sessionList = new ArrayList<AgentSession>();
createSnmP();
for (Agent agent : dbcon.readAgents()) {
    //XXX: TEST
    if (agent.isV3()) {
        snmp.getUSM().addUser(new OctetString(agent.getHostname()), agent.getUserModel());
    }
    sessionList.add(new AgentSession(agent));
}

```

Abbildung 4: LoadJDBCDBDriver-Methode

5.1.3.1.3 CreateSnmP

createSnmP() erzeugt und modifiziert das grundlegende Snmp-Objekt, ohne dem keine SNMP-Nachrichten, PDUs genannt (Protocol Data Units), verschickt werden können.

TransportMappings bestimmen welche Protokolle auf dem Transport- und Network-Layer des OSI-Modells benutzt werden. In diesem Fall wird das, laut RFC-3417, Standard-TransportMapping UDP über IPv4 verwendet. [IET]

Die MessageDispatcher behandeln das Versenden von PDUs und die Zuweisung der Antwort-PDUs zu bestimmten Agents. Dieser macht es möglich die Antworten der Geräte voneinander zu unterscheiden.

```

public void createSnmP() throws IOException {
    TransportMapping<?> transport = new DefaultUdpTransportMapping();
    snmp = new Snmp(transport);
    transport.listen();
    USM usm = new USM(SecurityProtocols.getInstance(), new OctetString(
        MPv3.createLocalEngineID()), 0);
    SecurityModels.getInstance().addSecurityModel(usm);
    MESSAGE_DISPATCHER.addMessageProcessingModel(new MPv2c());
    MESSAGE_DISPATCHER.addMessageProcessingModel(new MPv3(usm));
}

```

Abbildung 5: CreateSnmP-Methode

5.1.3.1.4 Process

Ist die *init*-Funktion erfolgreich durchlaufen, startet der Hauptprozess. Dieser regelt, in welchem Intervall die PDUs gesendet werden, und welchem Zyklus diese angehören.

Der *cycleTimeStamp* wird für das optionale Projektziel „Timeline“ benötigt. Mit diesem werden die Wellen an PDUs, deren Antworten die Daten für den Zustand des Netzwerkes zu einem Zeitpunkt liefern, eindeutig identifizierbar. So erlauben wir den Usern, vergangene Zustände des Netzes anzusehen.

```

public void process() throws IOException, InterruptedException,
    ParseException {
    try {
        while (true) {
            long cycleTimeStamp = new Date().getTime();
            for (AgentSession session : sessionList) {
                session.sendGet(snmp, MESSAGE_DISPATCHER, cycleTimeStamp);
            }
            Thread.sleep(INTERVAL);
        }
    } finally {
        snmp.close();
    }
}

```

Abbildung 6: Process-Methode

5.1.3.2 DatabaseConnection

5.1.3.2.1 Allgemeines

Die Klasse „DatabaseConnection“ regelt die benötigten Datenbank-Operationen, also die Schnittstelle zwischen SNMP-Manager, Animation und Benutzer. Dazu gehören das Einlesen der Liste der zu messenden Geräten (Agents) und das Schreiben der berechneten (Bandbreitenauslastung) und ausgelesenen (Hostname) Werte.

Wie in Kapitel 5.1.3.1.2 beschrieben, wird im Initialisierungsprozess die Methode *readAgents()* aufgerufen.

5.1.3.2.2 ReadAgents

Um die auszulesenden Agents zu identifizieren, wird ein gewöhnliches SQL-Select-Statement ausgeführt. Laut diesem werden folgende Eigenschaften zur Identifikation herangezogen: Die IP-Adresse, der Hostname, die Versionsnummer und der SNMPv3-Typ des Agents (siehe 4.8.3).

```

public Collection<Agent> readAgents() throws SQLException, UnknownHostException {
    PreparedStatement selectAgents = connection.prepareStatement(
        "select dev_k_devIP_id, dev_name, dev_version, dev_auth, dev_priv from `dev` order by dev_k_devIP_id",
        ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);

    Collection<Agent> agentCollection = new ArrayList<Agent>();
    ResultSet result = selectAgents.executeQuery();
    while (result.next()) {
        fetchAgents(agentCollection, result);
    }
    return fetchPorts((ArrayList<Agent>) agentCollection);
}

```

Abbildung 7: ReadAgents-Methode

5.1.3.2.3 FetchAgents

Die Funktion *fetchAgents()* beinhaltet die Prozedur zur Einteilung und Erstellung der Agents. Jedes Tupel der dev-Tabelle durchläuft diese Methode einmal.

```

private void fetchAgents(Collection<Agent> agentCollection, ResultSet result)
    throws SQLException {
    Address address = GenericAddress.parse(("udp:"
        + result.getString("dev_k_devIP_id") + "/" + Agent.UDP_PORT));
    String version = result.getString("dev_version");
    String hostname = result.getString("dev_name");
    String authPassword = result.getString("dev_auth");
    String privPassword = result.getString("dev_priv");
    if (version.equals("authPriv")) {
        agentCollection.add(new Agent(address, hostname,
            createUsmUser(hostname, AuthMD5.ID, authPassword, PrivDES.ID, privPassword), SecurityLevel.AUTH_PRIV));
        LOGGER.debug(String.format("Created v3-authPriv-Agent %s(%s)", hostname, address));
    } else if (version.equals("authNoPriv")) {
        agentCollection.add(new Agent(address, hostname,
            createUsmUser(hostname, AuthMD5.ID, authPassword, null, null), SecurityLevel.AUTH_NOPRIV));
        LOGGER.debug(String.format("Created v3-authNoPriv-Agent %s(%s)", hostname, address));
    } else if (version.equals("v2c")) {
        agentCollection.add(new Agent(address, hostname));
        LOGGER.debug(String.format("Created v2c-Agent %s(%s)", hostname, address));
    }
}

```

Abbildung 8: FetchAgents-Methode

5.1.3.2.4 FetchPorts

Sind alle Agents gefertigt, wird *fetchPorts()* aufgerufen. Darin werden den Agents die zugehörigen Ports zugewiesen. Die Zuweisungen ergeben sich aus der *if*-Tabelle, und erfolgen durch die Methode *add()* der Agent-Klasse.

```

private Collection<Agent> fetchPorts(ArrayList<Agent> agentList) throws SQLException {
    PreparedStatement selectIFs = connection.prepareStatement(
        "select if_k_devIP_id, if_portNumber from `if` order by if_k_devIP_id",
        ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);

    ResultSet result = selectIFs.executeQuery();
    while (result.next()) {
        String address = result.getString("if_k_devIP_id");
        int port = result.getInt("if_portNumber");
        for (Agent agent : agentList) {
            if (agent.getAddressString().equals(address)) {
                agent.addPort(port);
                LOGGER.debug(String.format("Added port #%d to Agent %s", port, agent.toString()));
            }
        }
    }
    return agentList;
}

```

Abbildung 9: FetchPorts-Methode

Hiernach ist das Programm bereit den Hauptprozess *process()* zu starten.

5.1.3.3 AgentSession

5.1.3.3.1 Allgemeines

Diese Klasse beinhaltet das Herzstück des SNMP-Managers. Sie organisiert im Wesentlichen das Verschicken und Empfangen der SNMP-Pakete und das Berechnen/Messen der Bandbreitenauslastung.

Eine Session ist „[...] eine logische Verbindung zwischen zwei adressierbaren Einheiten in einem Leitungsnetz, um Daten auszutauschen“. [WIK]

Folgende Instanzvariablen erfüllen die Zwecke:

- *agent*: Der Agent, also der gegenüberliegende Knoten dieser Verbindung
- *currentResponsePDU*: Die letzte empfangene PDU
- *responseListener*: Listener der die asynchrone Verarbeitung von PDUs ermöglicht
- *deltaTime*: Die Zeit zwischen dem Absenden einer PDU und dem Empfangen der Antwort-PDU
- *cycleTimeStamp*: Wie in 5.1.3.1.4 beschrieben
- *NetViewPDUFactory*: Klasse zur Erstellung und Konfiguration von PDUs (Kapitel 5.1.3.4)

```
public Agent agent;
private PDU currentResponsePDU;
private ResponseListener responseListener;
private long lastResponseTimestamp;
private double deltaTime;
private long cycleTimeStamp;
private static final NetViewPDUFactory NETVIEW_PDU_FACTORY = new NetViewPDUFactory();
```

Abbildung 10: Attribute der Agent-Klasse

5.1.3.3.2 SendGet

Die Funktion *sendGet()* wird vom Hauptprozess in NetViewMain periodisch aufgerufen. Durch diese wird der *cycleTimeStamp* übergeben, die *RequestId* erzeugt und die PDU geschickt.

Die RequestID, wird vom MessageDispatcher (siehe 5.1.3.1.3) erzeugt und sorgt für die Zuordnung von Antwort-PDU zu vorhergegangener Request-PDU.

```
public void sendGet(Snmp snmp, MessageDispatcher messageDispatcher,
    long cycleTimeStamp) throws IOException, ParseException {
    this.cycleTimeStamp = cycleTimeStamp;
    Integer32 requestId = new Integer32(
        messageDispatcher.getNextRequestID());

    PDU pdu = NETVIEW_PDU_FACTORY.createNetViewPDU(requestId, agent);
    snmp.get(pdu, agent.getTarget(), agent.getAddress(), responseListener);
    LOGGER.debug("Sending request to Agent:" + agent.getAddress());
}
```

Abbildung 11: SendGet-Methode

Jedes *AgentSession*-Objekt bekommt im Konstruktor einen *ResponseListener* zugeteilt. Diese werden benötigt um das asynchrone Handling von SNMP-GETs und deren Antworten zu erreichen.

Wird ein GET-Request abgesendet, muss der Listener im Aufruf angegeben werden. Nach dem Absenden wird dieser aktiv und arbeitet folgenderweise:

5.1.3.3.3 OnResponse

```
public void onResponse(ResponseEvent responseEvent) {
    if (responseEvent != null) {
        LOGGER.debug("Got response from agent:"
            + responseEvent.getPeerAddress());
        long time = new Date().getTime();
        setDelta(time - lastResponseTimestamp);
        setLastResponseTimestamp(time);
        currentResponsePDU = responseEvent.getResponse();
        if (currentResponsePDU != null) {
            int errorStatus = currentResponsePDU.getErrorStatus();
            if (errorStatus == PDU.noError) {
                LOGGER.debug(String.format(
                    "Response from agent:%s: %s",
                    responseEvent.getPeerAddress(),
                    currentResponsePDU.getVariableBindings()));
                handleBandwidthData();
            }
        } else {
            LOGGER.error("Error: Response PDU is null");
        }
    }
}
```

Abbildung 12: OnResponse-Methode 1

Die einzige Methode der anonymen, inneren Klasse *ResponseListener* ist *onResponse()*. In dieser findet sich eine Kette an if- und else-Abfragen die bestimmen wie die Antwort interpretiert wird.

Wird eine Antwort empfangen und kann einem *ResponseListener*-Objekt zugeordnet werden, wird der *responseEvent*-Variable ein gleichnamiges Objekt zugeordnet. Oder die Variable wird, wenn der Request in der vorher fixierten Zeit (*Agent.TIMEOUT*) unbeantwortet bleibt, *null*.

Läuft der Datenaustausch problemlos ab, wird zuallererst (erste if-Anweisung) Δt , und *lastResponseTimeStamp* zur Zeitmessung im nächsten Zyklus gesetzt.

Da eine Antwort jedoch auch zur Fehlerbescheinigung ankommen kann, muss die PDU überprüft werden. Diese wird über das *responseEvent* übergeben.

```

/*
 * without this, the library causes the program to receive
 * null-PDUs after responses are received (only async)
 */
((Snmp) responseEvent.getSource()).cancel(
    responseEvent.getRequest(), this);
} else {
    currentResponsePDU = null;
    handleResponsePDU();
    LOGGER.info(String.format("Agent %s timed out", agent.toString()));
}

```

Abbildung 13: OnResponse-Methode 2

Nach dem erfolgreichen Empfang einer Antwort, steht eine Anweisung, die verhindert dass danach null-PDUs empfangen werden. Darauf wird in der Dokumentation von SNMP4J hingewiesen.

Am Ende der if-Kette wird `currentResponsePDU` null zugewiesen um einwandfrei durch `handleResponsePDU()` zu durchlaufen.

5.1.3.4 HandleResponsePDU

In dieser Methode werden zufriedenstellend empfangene PDUs abgewickelt. Es werden die Werte `ifInOctets`, `ifOutOctets` und der Hostname aus dem Datagramm extrahiert.

Da pro Host mehrere Interfaces abgefragt werden können, müssen die *Variable Bindings* den zugehörigen AgentPorts zugeordnet werden. Dann wird der Hostname in die dev-Tabelle eingefügt und die Bandbreitenauslastungen in die if-Tabelle.

Von hier aus stehen die Daten der Animation zur Verfügung. Ein Zyklus des SNMP-Managers wurde vollständig durchlaufen.

```

private void handleResponsePDU() {
    DatabaseConnection dbcon = null;
    try {
        dbcon = DatabaseConnection.getInstance();
        for (AgentPort agentPort : agent.getPorts()) {
            int inBandwidthUtilization;
            int outBandwidthUtilization;
            if (currentResponsePDU != null) {
                inBandwidthUtilization = calculateBandwidthUtilization(
                    agentPort, OidPrefix.IF_IN_OCTETS);
                outBandwidthUtilization = calculateBandwidthUtilization(
                    agentPort, OidPrefix.IF_OUT_OCTETS);
                setAgentPortLastIfValuesFromCurrentResponsePDU(agentPort);
                dbcon.writeAgentProperties(agent, getVariableBindingVariable(null, OidPrefix.HOSTNAME).toString());
            } else {
                inBandwidthUtilization = 0;
                outBandwidthUtilization = 0;
                agentPort.setLastIfInOctetsValue(0);
                agentPort.setLastIfOutOctetsValue(0);
                agentPort.setLastIfSpeedValue(0);
            }
            dbcon.writeBandwidth(agent, agentPort.getPortNumber(),
                inBandwidthUtilization, outBandwidthUtilization,
                cycleTimeStamp);
        }
    } catch (ClassNotFoundException | SQLException e) {
        LOGGER.error(e.getMessage());
    } finally {
        if (dbcon != null) {
            try {
                dbcon.close();
            } catch (SQLException ex) {
            }
        }
    }
}

```

Abbildung 14: HandleResponsePDU-Methode

5.1.3.5 NetViewPDUFactory

Um das Erzeugen der Get-PDUs zu erleichtern, wurde eine Factory-Klasse geschrieben. Die einzige Methode der Klasse liefert eine PDU zurück. Durch den *agentPort*-Parameter werden die PDUs an Agents angepasst. Hierfür werden die Version und die Interfaces der Agents berücksichtigt.

```

public PDU createNetViewPDU(Integer32 requestId, Agent agent) throws ParseException {
    PDU pdu = new PDU();
    if (agent.isV3())
        pdu = new ScopedPDU();
    for (AgentPort agentPort : agent.getPorts()) {
        for (String oidPrefix : OidPrefix.getAllPerPortOidPrefixes()) {
            pdu.add(new VariableBinding(new OID(oidPrefix + "."
                + agentPort.getPortNumber())));
        }
    }
    for (String oidPrefix : OidPrefix.getAllPerAgentOidPrefixes()) {
        pdu.add(new VariableBinding(new OID(oidPrefix)));
    }
    pdu.setType(PDU.GET);
    pdu.setRequestID(requestId);
    return pdu;
}

```

Abbildung 15: CreateNetViewPDU-Methode

5.1.3.6 OidPrefix

Die, über SNMP abgefragten Attribute in der MIB, werden durch „Object Identifier“ identifiziert. OIDs bestehen aus durch Punkten getrennte Zahlen (z. B. „1.3.6.1.2.1.2.2.1.10“ für IfInOctets). Die sind hierarchisch geordnet. Jede OID kann als Baum dargestellt werden.

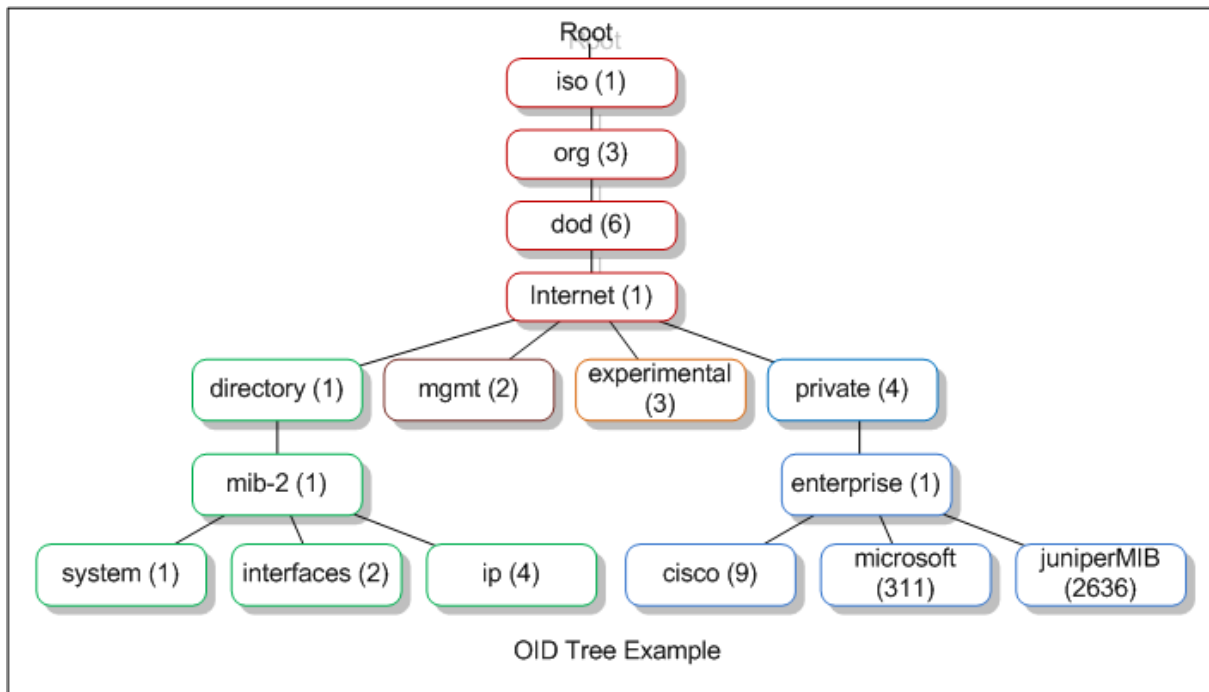


Abbildung 16: OID-Baum [OID]

Da OIDs an mehreren Stellen des Programms verwendet werden, wurde ein Enum-Typ erstellt. Enums oder Aufzählungstypen sind eine Sammlung von Konstanten. Diese können wie Datentypen verwendet werden.

Dadurch wird die Flexibilität des Codes erhöht. Sollen zu einem späteren Zeitpunkt zusätzliche Objekte mit SNMP behandelt werden, kann OidPrefix um diese erweitert werden.

```
public enum OidPrefix {
    IF_IN_OCTETS(".1.3.6.1.2.1.2.2.1.10"),
    IF_OUT_OCTETS(".1.3.6.1.2.1.2.2.1.16"),
    IF_SPEED(".1.3.6.1.2.1.2.2.1.5"),
    HOSTNAME(".1.3.6.1.4.1.9.2.1.3");

    private String oidPrefix;
```

Abbildung 17

5.2 3D-Umgebung, Web-Interface

5.2.1 Szenenaufbau in Three.JS

Die ersten Informationen, die für eine Three.JS-Anwendung benötigt werden, beziehen sich auf die Leinwand. Als solche dient ein simples HTML5 Canvas-Objekt, das entweder bereits im HTML-Dokument existiert oder via JavaScript erstellt und dem Dokument hinzugefügt wird. Als Leinwand festgelegt wird das Objekt durch seine ID.

Im folgenden Code werden eine Szene, eine Kamera und ein Renderer für die Leinwand erstellt:

```
// Init
var scene = new THREE.Scene();
var camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 1, 1000);
var renderer = new THREE.WebGLRenderer({ antialias: true });
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setClearColor(0xffffff, 1);
```

Abbildung 18: Funktion zum Erstellen der Leinwand

Größe und Hintergrundfarbe für die Leinwand sowie das Sichtfeld der Kamera werden definiert. Der Renderer ist der Teil des Programmes, der für die notwendigen Berechnungen von zum Beispiel Licht, Farben oder Positionen zuständig ist und schlussendlich den dreidimensionalen Raum als ein zweidimensionales Bild auf dem Bildschirm anzeigt.

5.2.2 Kamera

Three.JS stellt zwei Kameraarten zur Verfügung: Eine davon ist die sogenannte perspektivische Kamera, die Verzerrungen der Objekte durch Entfernung und Blickwinkel berücksichtigt (Perspektive) und dadurch realistischer wirkt. Die andere ist die orthographische Kamera, bei der keine Verzerrungen auftreten, sie stellt Objekte mathematisch korrekt dar. NetView verwendet eine perspektivische Kamera.

5.2.2.1 Kameras im Vergleich

Bei der perspektivischen Kamera (PerspectiveCamera) werden folgende Werte benötigt:

```
var camera = new THREE.PerspectiveCamera (
    75, window.innerWidth / window.innerHeight, 0.1, 1000
);
```

Abbildung 19: Erstellen einer perspektivischen Kamera

- fov - Field of View, Blickwinkel
- aspect - Seitenverhältnis von Höhe und Breite
- near - die nächste Ebene die berechnet wird
- far - die weiteste Ebene die berechnet wird

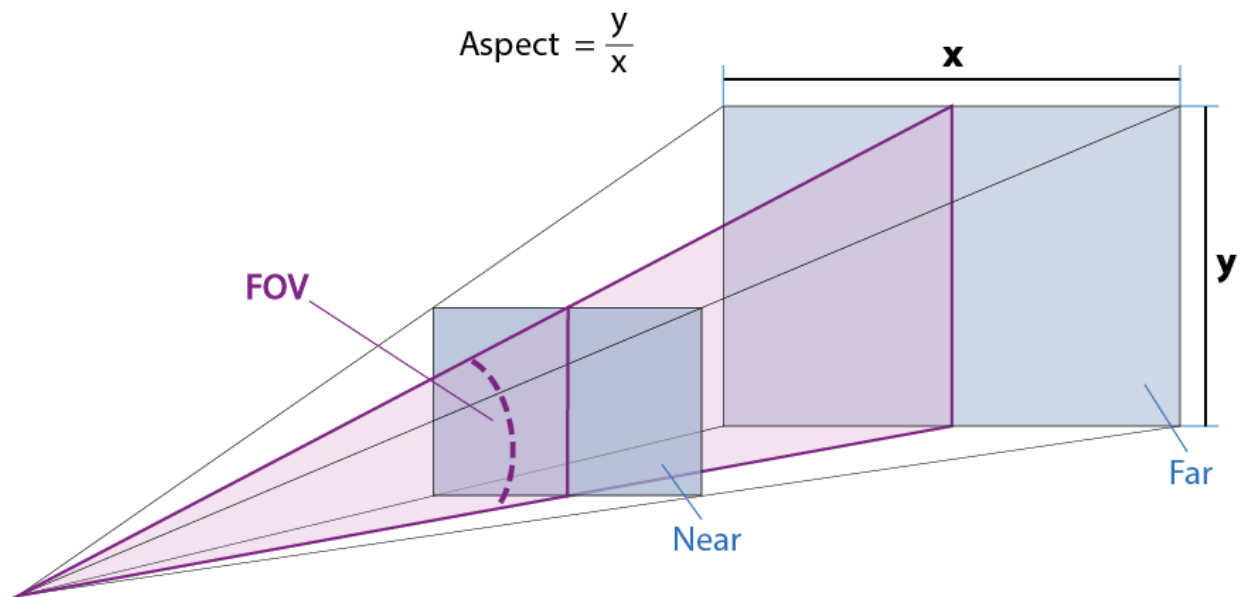


Abbildung 20: Blickfeld einer perspektivischen Kamera

Eine orthographische Kamera (OrthographicCamera) wird wie folgt erstellt:

```
var camera = new THREE.OrthographicCamera (
    width / - 2, width / 2, height / 2, height / - 2, 1, 1000
);
```

Abbildung 21: Erstellen einer orthographischen Kamera

- left - linke Begrenzungsebene
- right - rechte Begrenzungsebene
- top - obere Begrenzungsebene
- bottom - untere Begrenzungsebene
- near - die nächste Ebene die berechnet wird

- far - die weiteste Ebene die berechnet wird

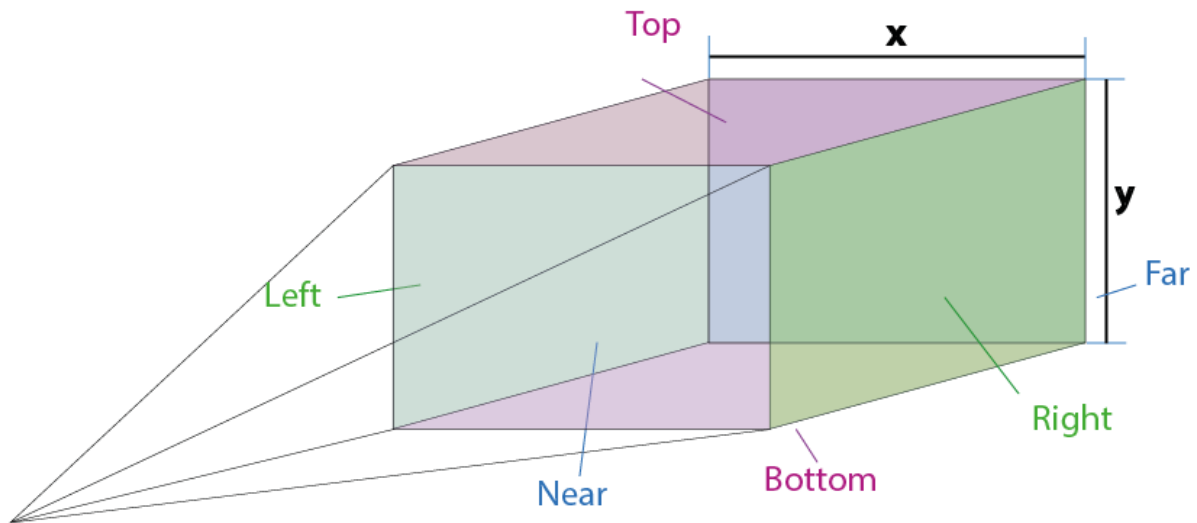


Abbildung 22: Blickfeld einer orthographischen Kamera

5.2.3 Modelle

5.2.3.1 Gebäude

Das Modell des Schulgebäudes wurde mit Hilfe grober Skizzen angefertigt. Es ist also nur eine ungefähre Nachbildung. Als Vorlagen dienten die Lagepläne, die an den Treppen in jedem Stockwerk im Gebäude zu finden sind. Da die einzelnen Pläne nicht sehr genau sind und im Größenverhältnis auch nicht zusammenpassen, musste ein einheitlicher Plan skizziert werden. Die einfachste und beste Methode für den Plan und auch für die spätere Umsetzung war das Skizzieren auf kariertem Papier. So sind mehrere Seiten entstanden, die mit Raumnummern versehen und digitalisiert wurden.

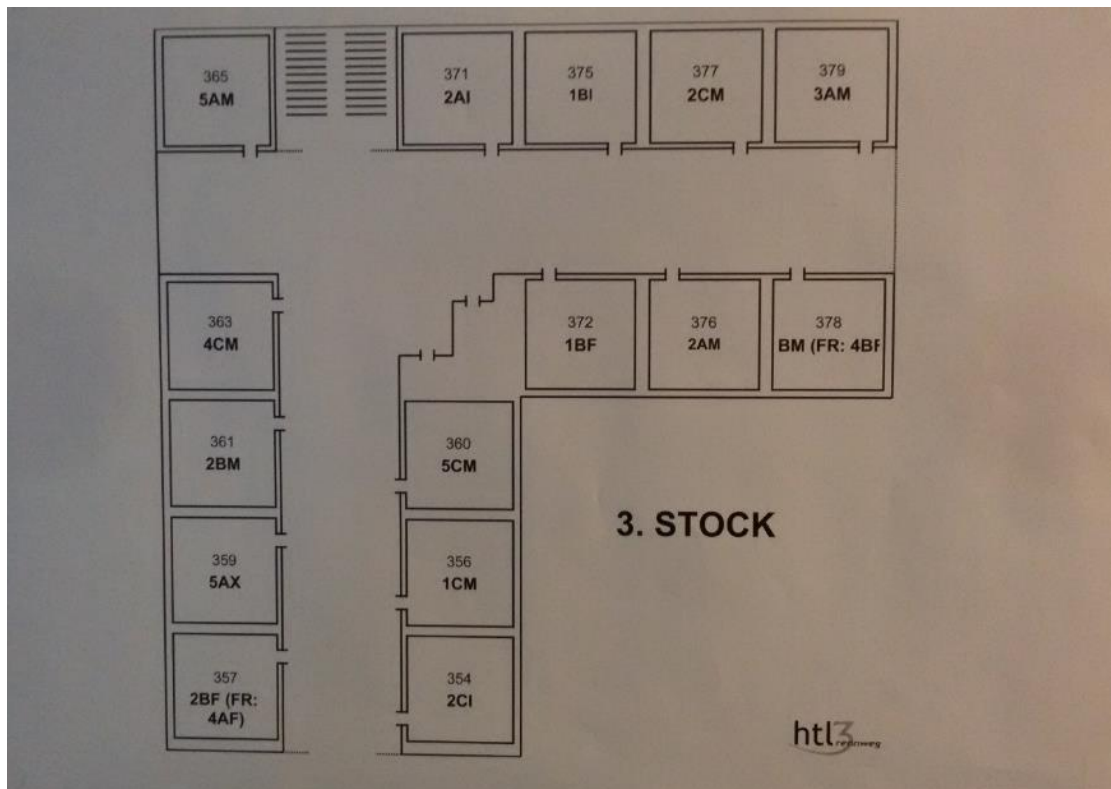


Abbildung 24: Raum Plan 3. Stock (Foto des Originals)



Abbildung 23: Digitalisierte Version der Pläne mit Raumnummern

Modelliert wurden die einzelnen Ebenen direkt in Three.js. Jede modellierte Etage besteht aus einer individuell definierten Geometrie, bestehend aus zweihundert bis vierhundert miteinander verbundenen Eckpunkten, von denen jeder einzelne per Hand definiert werden musste. Zusätzlich wurden zur besseren Übersicht auch Böden eingefügt, dargestellt durch einzelne, aneinanderliegende, zweidimensionale Rechtecke, den sogenannten Planes. In Summe ergeben sich ungefähr 1700 verschiedene Koordinatenwerte, die zur Berechnung des Gebäudemodells benötigt werden.

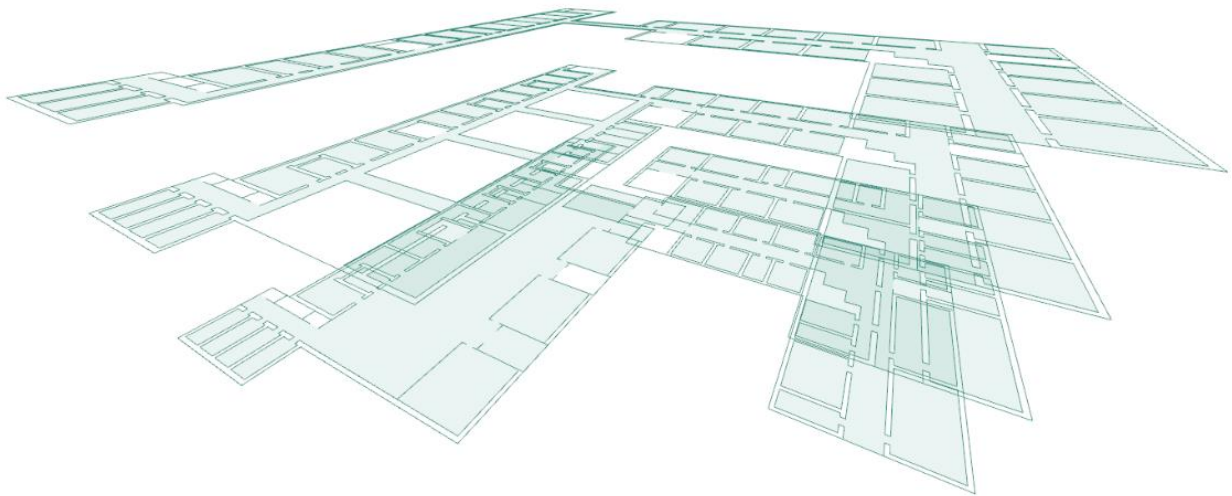


Abbildung 25: Leeres Gebäudemodell im 3D-Interface

5.2.3.2 Geräte

Die Modelle der Netzwerkgeräte wurden in Blender erstellt und texturiert. Dabei wurden für Router und Switch in einfache Grundformen, wie Zylinder und Würfel, mit dem Knife-Tool Linien geschnitten und die dadurch entstandene Form abgesenkt.

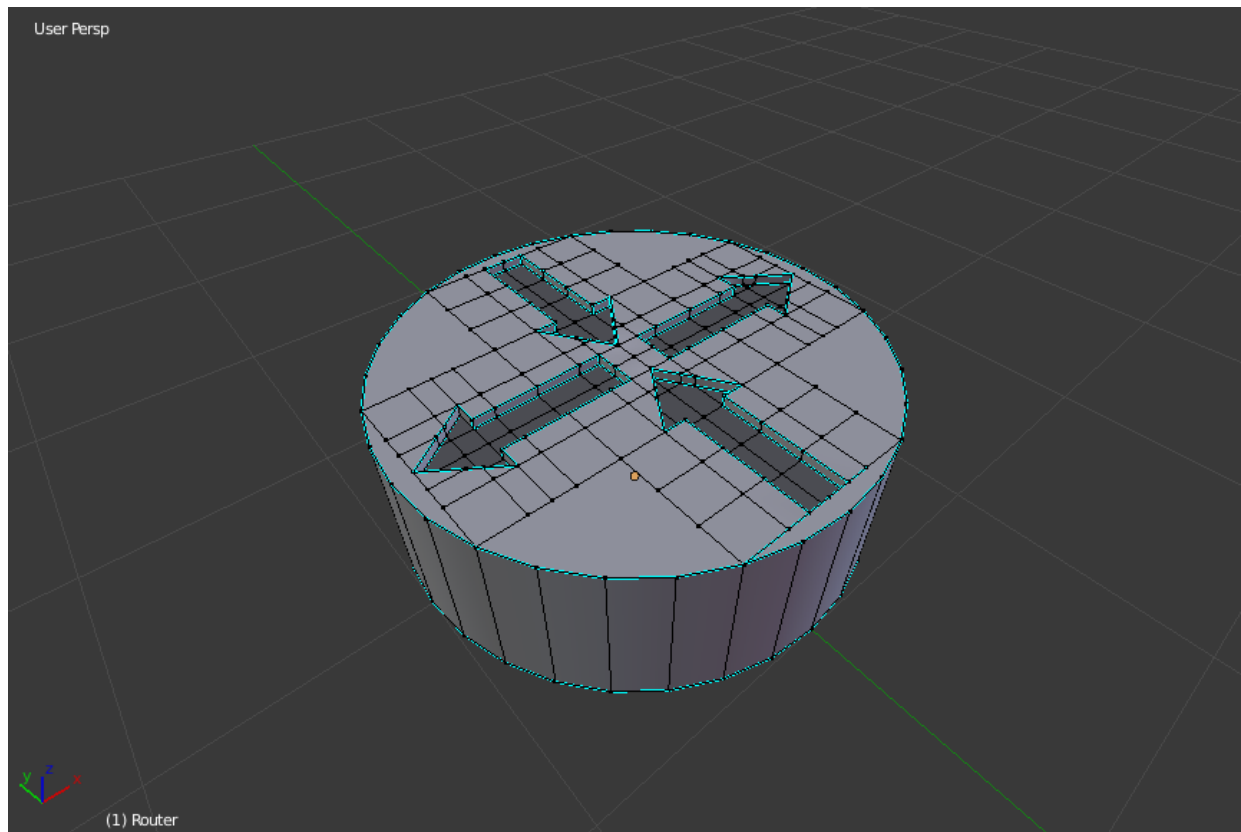


Abbildung 26: Entstehung des Router-Modells

Beim Modellieren des Accesspoints wurde eine einfache Kugel auf einen verformten Zylinder gesetzt, um eine typische Antennenform zu erzielen. In Kombination mit durch Three.js animierten Wellen soll ein W-LAN Netz dargestellt werden.

Alle Gerätemodelle wurden texturiert und als JSON-Datei exportiert. Der Exporter für Blender ist im Downloadpaket von Three.js enthalten. Da Three.js Licht aus Performancegründen nicht korrekt berechnet, sondern nur ungefähr imitiert, sehen die Texturen der Modelle im 3D-Interface ein wenig anders aus als in Blender gerendert.



Abbildung 27: Zwei der fertigen Modelle

5.2.4 Antialiasing

Antialiasing ist eine Methode zur Glättung von Kanten in Computergrafiken. Dabei werden die Farben eines Bildpunktes nicht nur anhand der Position des Bildpunktes sondern auch anhand der Position im dreidimensionalen Raum und der Farben der umgebenden Bildpunkte berechnet. Antialiasing vermindert zum Beispiel den sogenannten Treppeneffekt, der durch die Anordnung der Bildpunkte eines Bildschirms entsteht. Da die einzelnen Bildpunkte, auch Pixel genannt, rasterähnlich angeordnet sind, kann eine diagonale Linie nicht als gerade Linie angezeigt werden, sondern nur als diagonal aneinanderliegende Pixel. Dadurch, dass Pixel ungefähr quadratisch sind, entsteht unter anderem der bereits genannte Treppeneffekt, der bei unseren Modellen durch viele einzelne Linien im dreidimensionalen Raum häufig auftritt.

5.2.5 Werte

Jede Leitung in unserem Modell wird durch eine einfache, farbige Linie dargestellt, die immer genau zwei Netzwerkgeräte verbindet. Die Farbe einer Linie ist abhängig von der Auslastung der repräsentierten Leitung. Der Wert der Auslastung ist in Prozent angegeben. Er wird berechnet aus dem Durchschnitt von Input und Output, ausgehend von einem der beiden durch diese Leitung verbundenen Geräte. In regelmäßigen Abständen wird dieser Wert aus der Datenbank ausgelesen und die Farbe der zugehörigen Linie aktualisiert.

5.2.6 Aufbau des dreidimensionalen Raums

5.2.6.1 Datenübermittlung

Eines der Ziele bei der Umsetzung war, möglichst alle erforderlichen Daten zur korrekten Darstellung des kompletten Modells in der Datenbank zu halten. Aus diesem Grund wird beim Laden des Interfaces eine Abfrage nach vorhandenen Geräten und Verbindungen an die Datenbank gesendet. Die Antwort kommt in Text-Form und wird als JSON, als JavaScript Objekt, interpretiert. So können alle Daten wie bei einem Array mit Leichtigkeit ausgelesen und weiterverarbeitet werden.

```
{ "geraete_id":"356_0" , "ziel_id":"300_0" , "device":"Accesspoint" , "in":34 , "out":34 }
```

Abbildung 28: Beispiel eines Datensatzes im JSON-Format

Mit Hilfe dieser Daten wird für jedes eingetragene Gerät die Position aus einer vorher definierten Liste gesucht und das entsprechende 3D-Modell platziert.

Die Abfrage an die Datenbank wird in einem PHP-File ausgeführt, welches wiederum via AJAX-Call¹ von Prototype.JS im Hintergrund aufgerufen wird. Die Antwort der Datenbank wird zu Text weiterverarbeitet und zurück an den JavaScript-Aufruf gesendet, wo die Informationen als JSON interpretiert wie zuvor beschrieben weiterverarbeitet werden.

Im PHP-File sind Adresse und Zugangsdaten der Datenbank sowie die SQL-Abfrage definiert. Die Weiterverarbeitung zu einem Text (String) erfolgt direkt nach der Abfrage, dazu werden die Antworten aus einem Array ausgelesen und wie im Codebeispiel aneinandergehängt:

```
<?php

$db = mysql_select_db(
    'netviewdatabase',
    mysql_connect( '10.0.0.100:3306' , 'netview' , 'NetView123!' )
);

$query = mysql_query( "SELECT * FROM `if`" );

$text = "";

while($row = mysql_fetch_array($query)){
    $text = $text.'{"geraete_id":"' . $row[1]
        . '","ziel_id":"' . $row[2]
        . '","device":"' . $row[3]
        . '","in":"' . $row[4]
        . '","out":"' . $row[5] . '}',';
}

$text = substr($text, 0, -1);

echo "[".$text."]";

?>
```

Abbildung 29: PHP-File mit Datenbankabfrage

Jede Zeile aus der Datenbank wird in diese Form gebracht und an den String angehängt. Nach der letzten Zeile wird das letzte Zeichen "," entfernt, da beim Interpretieren des Strings als JSON ein Fehler auftreten würde, weil der Compiler nach einem Komma nach einem weiteren Datensatz sucht.

Eine asynchrone JavaScript-Funktion, auch AJAX-Call genannt, gibt den Befehl zum Ausführen des PHP-Codes und verarbeitet, falls bei der Übertragung kein Fehler auftritt, das Ergebnis. Dabei wird durch die ID für jede Zeile festgestellt, um welches Gerät es sich handelt und wo es sich befindet. Mit den Ergebnissen wird das entsprechende Modell erstellt und positioniert. Unmittelbar danach wird mit Hilfe der Geräte-ID und der Ziel-ID die Leitung zwischen beiden Positionen gezogen. Diese hat – da noch keinerlei Informationen über die

¹ AJAX-Call steht für asynchroner JavaScript Aufruf, eine JavaScript Funktion die im Hintergrund des laufenden Programmes ausgeführt wird.

aktuelle Auslastung abgefragt wurden – die Farbe Schwarz, eine Auslastung von null Prozent.

```
new Ajax.Request('getData.php', {
  method: 'get',
  onSuccess: function(transport) {
    var response = transport.responseText || "no response text";
    var stats = JSON.parse(response);
    for(i = 0; i < stats.length; i++){
      if(xyz[stats[i].geraete_id] != undefined){
        createDev(xyz[stats[i].geraete_id], stats[i].device, stats[i].geraete_id);
        if(line[stats[i].geraete_id+"-"+stats[i].ziel_id] != undefined){
          scene.add(line[stats[i].geraete_id+"-"+stats[i].ziel_id]);
        }
      }
    }
  },
  onFailure: function() { alert('Something went wrong...'); }
});
```

Abbildung 30: Aufruf des PHP-Files beim Laden der Seite

Eine ähnliche Funktion wird zum Aktualisieren der Daten verwendet. Wieder wird PHP-Code ausgeführt, um Daten aus der Datenbank zu erhalten. Der Unterschied zum erstmaligen Laden ist, dass diese Funktion sich wiederholt. In der Theorie sollte sie ein Mal pro Sekunde ausgeführt werden. In der Praxis hat sich herausgestellt, dass diese Sekunde erst mit Abschluss der vorherigen Ausführung gezählt wird.

5.2.6.2 Anordnung, Positionierung, Erweiterbarkeit

Um eine geordnete Struktur zu schaffen und gleichzeitig Erweiterbarkeit zu garantieren, musste jede mögliche Position und Leitung vordefiniert werden. Ausschlaggebend für die Anordnung der Modelle sind die Raumnummern im Gebäude. Jedem Raum ist in seiner Mitte ein Punkt zugeordnet, an dem, falls vorhanden, das jeweilige Gerät positioniert wird. Für jeden Raum gibt es vier solcher Punkte in vier verschiedenen Höhen. Alle diese Punkte sind in einem Array definiert. Somit ist es ohne Codeänderungen möglich, in jedem Raum bis zu vier Geräte in die Datenbank einzutragen und darstellen zu lassen. Identifiziert werden sie durch Raumnummer und Position. Dabei wird die unterste Position zuerst gezählt, begonnen wird bei null. Wenn sich im Raum 356 zum Beispiel ein Switch befindet, an dem ein Accesspoint angeschlossen ist, würde der Switch die ID 356_0 und der Accesspoint die ID 356_1 erhalten. Zusätzlich zu den Räumen sind noch Positionen für Verteilerpunkte, zum Beispiel für die jeweilige Etage, definiert.

Bei den Leitungen herrscht ein ähnliches System wie bei den Geräten. Da theoretisch jedes Gerät mit jedem verbunden sein kann, gibt es auch eine deutlich höhere Anzahl an vordefinierten Leitungen als an Gerätepositionen. In der Praxis wurde das Leitungssystem aber soweit reduziert, dass es nur möglich ist, alle Geräte in einem Raum miteinander zu verbinden und jedes einzelne dieser Geräte mit dem jeweiligen Etagenverteiler. Alleine dadurch entstehen mindestens sieben mögliche Leitungen für jede mögliche Geräteposition. Identi-

fiziert werden die Leitungen durch Start- und Endpunkt. Zusammen ergeben sie eine Bezeichnung wie zum Beispiel 356_1-300_0. Das wäre die Leitung zwischen dem zweiten Gerät im Raum 356 und dem ersten Gerät am Verteilerpunkt im 3. Stock.

5.2.6.3 Farbschema

Zur leichteren Feststellung des derzeitigen Wertes einer Leitung eignet sich am besten eine farbliche Markierung. Dazu wurde eine Farbtabelle angelegt, die von Rot – sehr hohe Auslastung – über Orange, Gelb und Grün bis zu Blau – sehr geringe Auslastung – reicht. Jeder Farbton repräsentiert ein Intervall von je zehn Prozent.



Abbildung 31: Farbtabelle

5.2.7 Übersichtlichkeit

5.2.7.1 Genaue Werte darstellen

Farben sind der einfachste Weg einen ungefähren Wert darzustellen, aber durch eine Färbung alleine kann kein genauer Wert festgestellt werden. Aus diesem Grund wird zu jeder Leitung der Wert auch als Zahl in einem sogenannten Sprite wiedergegeben. Sprites sind zweidimensionale Flächen, die immer in Richtung Kamera zeigen. Three.js hat in der aktuellen Version keine fertige Funktion für Textsprites, deshalb wurde bei der Umsetzung eine Grafik mit dem Wert als Hintergrund des Sprites verwendet. Diese Grafik wird dynamisch als Canvas-Objekt erstellt. In dieses Canvas-Objekt kann beliebiger Text – in diesem Fall die Auslastung der Leitung in Prozent – eingefügt und formatiert werden. Ist man mit dem Ergebnis zufrieden, sendet man das Canvas-Objekt als Grafik an das Sprite-Objekt, welches die Grafik wiederum als Hintergrund anzeigt.

< Platzhalter >

Abb. x - Auslastung einer Leitung in Prozent

Da ein Canvas-Objekt eine Pixelgrafik beinhaltet, muss besonders auf die richtige Größe geachtet werden. Ebenso lässt sie sich nicht einfach verändern. Um einen anderen Wert anzeigen zu lassen, müssen ein neues Canvas-Objekt und ein neuer Sprite erstellt, sowie der bereits existierende Sprite gelöscht werden.

Aus diesem Grund und da bei der hohen Anzahl von Geräten im Netzwerk schnell die Übersicht verloren gehen kann, sind die genauen Werte als Zahl ausgeblendet. Links im Interface befindet sich eine Leiste mit Kontrollfeldern zum Ein- und Ausblenden der Sprite-Objekte für die jeweilige Etage.

5.2.7.2 Gerätebeschriftungen

Das gleiche Prinzip wie bei der Darstellung der genauen Auslastungswerte wird bei den individuellen Beschriftungen der Geräte angewendet. Allerdings wird hier nicht mit jeder Datenaktualisierung der Sprite gelöscht und neu erstellt sondern, wie die Netzwerkgeräte, einmalig beim Laden der Seite erstellt und positioniert. Die Gerätebeschriftungen bleiben standardmäßig ebenso wie die Auslastungswerte zur besseren Übersicht ausgeblendet.

5.2.7.3 Etagen

Nicht nur Beschriftungen können durch ihre jeweiligen Kontrollkästchen für die jeweilige Etage ein- und ausgeblendet werden, sondern auch ganze Stockwerke. Das betrifft die Raumkonturen, Geräte, Leitungen und optional eingeblendete Beschriftungen.

Besondere Probleme machten hierbei das Wiedereinblenden der Leitungen, da hierfür festgehalten werden musste, welche Leitungen überhaupt geladen wurden. Anders als bei den Geräten, bei denen die Modelle aus externen Dateien geladen werden, sind die Leitungen alle immer im Code. Verwendet man eine Funktion, die alle Leitungen dieser Etage ausblendet, spielt das keine Rolle, da nicht sichtbare Leitungen einfach übersprungen werden. Beim Einblenden wiederum können nicht einfach alle Leitungen gewählt werden, da sonst auch alle eigentlich nicht gewünschten Linien sichtbar werden. Die Lösung dieses Problems stellte ein Array dar, in das die Identifizierungen aller sichtbaren Leitungen bei deren Erstellung gespeichert werden. Beim Einblenden werden nun nur mehr jene Leitungen angesprochen, deren IDs in diesem Array gespeichert sind.

5.3 Timeline

Da die Umsetzung der Hauptziele bereits Mitte Februar, mit Ausnahme des Diplomarbeitbuches, fast vollständig abgeschlossen war, begannen mit März die ersten Arbeiten an der Timeline. Die Timeline ist ein optionales Feature, mit dem zu jeden beliebigen Zeitpunkt, der in der Datenbank gespeichert wurde, zurückgesprungen und die Auslastung des Netzes zu dieser Zeit beobachtet werden kann. Dafür ist es notwendig, alle jemals gemessenen Daten festzuhalten. Dazu existiert eine eigene Tabelle in der NetView-Datenbank (siehe 6 Datenbank).

Die Timeline wurde als eigenständige Seite als Gegensatz zum Livebetrieb erstellt. Umgesetzt wurde sie wieder mit Hilfe eines AJAX-Calls, wie dem, der beim Laden der Seite ausgeführt wird. Allerdings wird die JavaScript-Funktion nicht beim Laden oder, wie beim

Live-Modus, in regelmäßigen Abständen ausgeführt, sondern durch Änderungen am Wert eines Schiebereglers. Sobald der Regler bewegt wird, wird der AJAX-Call gesendet, der die Auslastungswerte zur ausgewählten Uhrzeit aus der Datenbank holt.

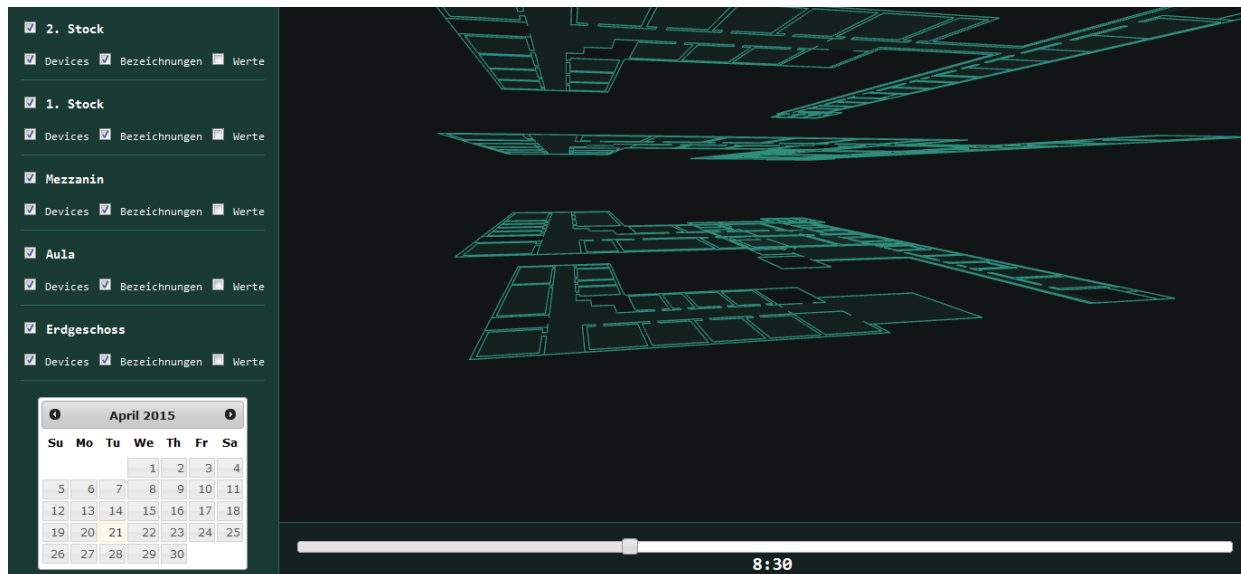


Abbildung 32: Datumswahl und Schieberegler der Timeline

Das Datum wird mit Hilfe von eines von jQuery bereitgestellten Kalenders eingestellt. Die Uhrzeit wird durch einen Schieberegler festgelegt. Der Schieberegler, auch kurz Slider genannt, wurde ebenfalls mit Hilfe der JavaScript Bibliothek jQuery erstellt. Er repräsentiert einen Tag mit vierundzwanzig Stunden und ist in Intervalle von zehn Minuten unterteilt.

```
// Slider
jQuery("#slider").slider({
    value: 1,
    range: "min",
    min: 0,
    max: 1440,
    step: 10,
    slide: function (event, ui) {
        slider_value = ui.value*60;
    }
});
```

Abbildung 33: Erstellen des Sliders

6 Datenbank

6.1 Anforderungen

Sowohl auf Seiten des Web-Interfaces als auch auf Seiten unseres SNMP Managers war es notwendig unterschiedlichste Daten, wie zum Beispiel Adressen und Standort der zu messenden Geräte zu speichern und auszulesen. Außerdem müssen sämtliche gemessenen Werte zwischengespeichert und mit den entsprechenden Geräteinformationen verknüpft werden um sie in unserem Modell anzeigen zu können. Ziel war es, die Daten möglichst effizient zu speichern, sowie die Anzahl und Komplexität unserer Tabellen möglichst gering zu halten. Um die Struktur unserer Datenbank zu planen haben wir ein ER-Modell entwickelt(siehe 6.2).

6.1.1 Wahl der Datenbanksoftware

Wir haben uns für einen MySQL-Datenbankserver entscheiden, da dieser sehr schnell, zuverlässig und einfach zu benutzen ist. MySQL Server hat darüber hinaus viele praktische Features, die in enger Zusammenarbeit mit den Benutzern entwickelt wurden.

MySQL Server wurde ursprünglich für den Zweck entwickelt, große Datenbanken sehr viel schneller als bestehende Lösungen zu handhaben. Er ist seit Jahren in äußerst anspruchsvollen Produktionsumgebungen im Einsatz. Obwohl er ständig weiterentwickelt wird, bietet MySQL Server schon heute ein reiches Set an Funktionen. Flexible Anbindungsmöglichkeiten, Geschwindigkeit und Sicherheit machen MySQL Server besonders geeignet für den Zugriff auf Datenbanken im Internet.

Große Vorteile bietet MySQL auch in der Kommunikation mit unserem Web Interface. Während bei einer herkömmlichen Datenbankabfrage mittels PHP mehrere Codezeilen erforderlich sind erfordert eine MySQL-DB nur eine einzige Zeile.

Ausschlaggebend für mich war, dass ich mit den Datenbankservern Oracle-DB und MicrosoftSQLServer bereits gearbeitet habe. Da ich mir im Rahmen der Diplomarbeit neues Wissen aneignen wollte habe ich mich für MySQL entschieden.

6.2 Modell

Es war uns wichtig unser Modell so übersichtlich wie möglich zu gestalten und nur die wichtigsten Informationen darin abzubilden. Das Datenbankmodell besteht aus den 3 miteinander Verknüpften Tabellen „device“(dev), „interface“(if) und „history“(h).

Zwischen den Spalten `dev_k_devIP_id` und `if_k_devIP_id` der Tabellen *dev* und *if* besteht eine 1:n Verbindung. Vom Primary-Key der Tabelle *if* der sich aus den Feldern `if_k_devIP_id` und `if_portNumber` zusammensetzt besteht eine 1:n Verbindung zur Tabelle *h* und deren Spalten `h_k_devIP_id` und `h_portNumber`.

Tabelle	Zusammensetzung des Primarykeys		
device	dev_k_devIP_id		
interface	if_k_devIP_id	if_portNumber	
history	h_k_devIP_id	h_portNumber	h_timestamp

Tabelle 3: Primarykeys

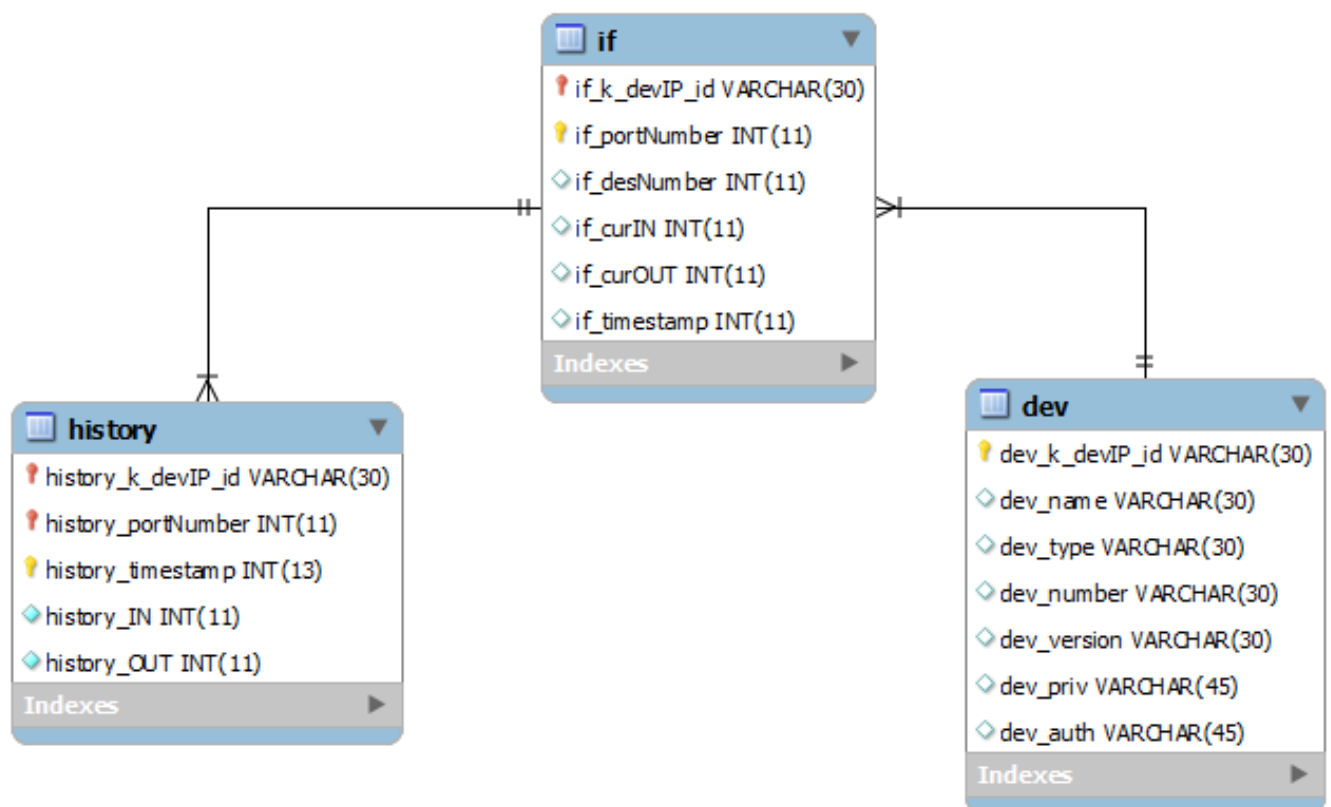


Abbildung 34: ER-Modell

6.3 Datenstruktur

Wie schon bereits erwähnt besteht unsere Datenbank aus 3 miteinander verknüpften Tabellen, jede Tabelle wird in unterschiedlichen Anwendungsbereichen verwendet. Um Abfragen senden zu können benötigt unser SNMP-Manager Werte aus den Tabellen „device“ und „interface“, genauer gesagt IP-Adresse, Interfacenummer, sowie die die SNMP-Konfiguration betreffenden Felder „version“, „priv“ und „auth“. IP-Adresse und Interfacenummer werden benötigt um eine Abfrage zu adressieren, da wir mehrere SNMP-Versionen unterstützen muss festgelegt werden auf welchem Gerät welche Version unterstützt wird. Außerdem muss je nach Version ein Key für Verhashung und Verschlüsselung vergeben werden.

Die Ergebnisse seiner Berechnungen speichert er in der Tabelle „interface“ in die Spalten „curIN“, „curOut“ und „timestamp“. Da eine Leitung einen Up- und einen Downstream hat und sich die Auslastung dieser stark unterscheiden kann, werden diese in unterschiedlichen Spalten gespeichert. Die restlichen 3 Felder der Tabelle „device“, „name“, „type“ und „number“ werden vom Web-Interface benötigt um die passenden Gerätesymbole an ihre richtige Position zu setzen.

Die Tabelle „history“ hat denselben Aufbau wie die Tabelle „interface“, der Unterschied ist, dass in der Tabelle „interface“ nur die jeweils aktuellen Messungen eingetragen sind. In der Tabelle „history“ hingegen werden die Einträge nicht überschrieben, sondern jeder gemessene Wert bleibt in der Tabelle vorhanden, um unserem zusätzlichen Feature der Timeline(siehe 5.3) zu ermöglichen den Verlauf der Messungen über einen längeren Zeitraum zu betrachten.

6.4 Datenbankeinträge

Wie man einen Eintrag in der Datenbank vornimmt, sowie eine Erklärung welche Werte eingetragen werden müssen finden Sie im Implementierungshandbuch.

Da einige Einträge von unserem SNMP Manager vorgenommen werden und daher kein Zutun des Administrators benötigen oder nur zur besseren Übersicht dienen, dürfen diese Felder den Wert Null haben.

Tabelle	NULL-Wert möglich		
device	name	priv (*)	auth (*)
interface	curIN	curOut	timestamp
history	Sämtliche Einträge werden automatisch getätigt		

Tabelle 4: NULL-Werte

*nur bei SNMPv2 oder v3 ohne Authentifizierung NULL-Wert erlaubt

Werte die zum Messen oder platzieren der Geräte benötigt werden müssen der Datenbank auf jeden Fall übergeben werden und dürfen den Wert Null daher nicht annehmen.

6.4.1 Notwendige Einträge

Um einen Eintrag in der Datenbank zu tätigen, kann man entweder die graphische Benutzeroberfläche verwenden oder man schreibt ein entsprechendes SQL-Script.

Sämtliche Einträge im SQL-Script haben in folgender Syntax zu erfolgen:

```
INSERT INTO `netviewdatabase`.`table_name`
(`column1_name`,
`column2_name`)
VALUES
(<{value1_name: value1}>,
<{value2_name: value2}>);
```

6.4.1.1 Tabelle dev

Für jedes zu messende Gerät muss eine zugehörige IP-Adresse eingetragen werden.

```
INSERT INTO `netviewdatabase`.`dev` (`dev_k_devIP_id`) VALUES ('10.0.0.3');
```

Außerdem soll der Gerätetyp in der 3. Spalte als „switch“, „router“ oder „firewall“ eingetragen werden.

```
INSERT INTO `netviewdatabase`.`dev` (`dev_type`) VALUES ('switch');
```

Der Standort eines Gerätes wird mittels Raumnummer bekannt gegeben, diese wird benötigt um es im Web-Interface entsprechend platzieren zu können. Mit einem Unterstrich getrennt muss angegeben werden, um das wievielte Gerät, innerhalb des Raumes, es sich handelt.

```
INSERT INTO `netviewdatabase`.`dev` (`dev_number`) VALUES ('310_0');
```

Da unsere Anwendung 3 verschiedene SNMP Versionen unterstützt, muss im Feld dev_version die jeweils verwendete Implementierung mit einem der folgenden Kürzel angegeben werden: noAuthPriv, authPriv, v2c.

```
INSERT INTO `netviewdatabase`.`dev` (`dev_version`) VALUES ('authPriv');
```

Da die beiden v3 Implementierungen authPriv und noAuthPriv Authentifizierung erfordern müssen in die Spalten dev_priv und dev_auth SNMP Passwörter für Hash und Verschlüsselung eingetragen werden.

```
INSERT INTO `netviewdatabase`.`dev` (`dev_priv`) VALUES ('password1');  
INSERT INTO `netviewdatabase`.`dev` (`dev_auth`) VALUES ('password2');
```

6.4.1.2 Tabelle if

Es muss für jedes zu messende Interface die IP-Adresse des in der Tabelle dev definierten Geräts eingetragen werden.

```
INSERT INTO `netviewdatabase`.`if` (`dev_k_devIP_id`) VALUES ('10.0.0.3');
```

Jedes Interface hat eine MIB in der alle Portnummern vermerkt sind, die Portnummer, des zu messenden Interfaces, muss in die Spalte if_portNumber eingetragen werden.

```
INSERT INTO `netviewdatabase`.`if` (`if_portNumber`) VALUES ('10001');
```

Um die Leitungen zwischen den Geräten dynamisch ziehen zu können wird das Feld if_desNumber benötigt. Hierbei handelt es sich um die Raumnummer, wo jenes Gerät steht zu dem die Leitung gezogen werden soll. Mit einem Unterstrich getrennt muss angegeben werden, um das wievielte Gerät, innerhalb des Zielraumes, es sich handelt.

```
INSERT INTO `netviewdatabase`.`if` (`if_desNumber`) VALUES ('305_0');
```

Beispiel:

Zwei Switches(A, B) mit den IP-Adressen 10.0.0.1(A) und 10.0.0.2(B) stehen in den Räumen 365(A) und 310(B). In Raum 310 steht bereits ein anderes Gerät. Sie sind beide am Interface fa0/1 miteinander verbunden. A unterstützt v3 authPriv, B nur v2c. Die in die Datenbank einzutragenden Werte haben folgendermaßen auszusehen:

dev_k_devIP_id	dev_name	dev_type	dev_number	dev_version	dev_priv	dev_auth
10.0.0.1	Switch365	switch	365_0	authPriv	passwort123!	netview123!
10.0.0.2	Switch310	switch	310_1	v2c	NULL	NULL

Abbildung 35: Beispiel dev-Tabelle

if_k_devIP_id	if_portNumber	if_desNumber	if_curIN	if_curOUT	if_timestamp
10.0.0.1	10001	310_1	NULL	NULL	NULL
10.0.0.2	10001	365_0	NULL	NULL	NULL

Abbildung 36: Beispiel if-Tabelle

6.5 Timestamp

Jedem Eintrag in den Tabellen interface und history wird mit der Übergabe einer Messung auch ein Timestamp übergeben, dadurch wird jeder Messung ein Datum und eine genaue Uhrzeit übergeben. NetView verwendet den sogenannten Unix Timestamp der wie folgt definiert wird:

Die Unixzeit ist eine einfache Zeitdefinition, die 1969 für das Betriebssystem Unix entwickelt und als POSIX-Standard festgelegt wurde. Seit Unix Version 6 zählt die Unixzeit die vergangenen Sekunden seit Donnerstag, dem 1. Januar 1970 00:00 Uhr UTC.

[UNIX]

Die Darstellung eines Datums in Sekunden wird häufig verwendet, da sie für Computerprogramme leichter zu verarbeiten ist als das menschliche Datumsformat. Zeiträume lassen sich mit diesen Werten besonders leicht berechnen, da man sie miteinander subtrahiert. Sommer- und Winterzeit spielen hier keine Rolle mehr, daher wird die Unixzeit meist als Zeitstempel verwendet. Aufgrund der oben genannten Vorteile wird der Unixzeitstempel meist für die Erzeugung und den Vergleich von zeitbezogenen Datenbankeinträgen verwendet. Ein weiterer positiver Aspekt ist die Möglichkeit zur einfachen Konvertierung in unser gewohntes Datumsformat und umgekehrt, zum Beispiel durch die vordefinierten PHP-Funktionen `date()` und `mktime()`.

Beispiel für einen Unix Timestamp:

1425462936

Mittwoch, 4. März 2015 10:55:36 UTC+1

Wir verwenden den Unix Timestamp, um jeder Messung des SNMP-Managers einen genauen Zeitpunkt zuzuweisen. Hierbei ist es besonders wichtig, dass jeder Messung eines Interfaces ein unterschiedlicher Timestamp zugeteilt wird, da dieser in der Tabelle „history“ als Teil des Primärschlüssels fungiert. Da wir allerdings zwischen zwei Messungen am selben Interface immer mindestens eine Sekunde lang keine Aktion setzen besteht hier keinerlei Gefahr, dass es hierbei zu einer Verletzung der Integrität kommt.

Da wir für jeden gemessenen Wert auch einen genauen Zeitpunkt festlegen, ist es möglich sich die Auslastung des Netzwerks rückwirkend, das bedeutet zu jedem beliebigen Zeitpunkt in der Vergangenheit, anzeigen zu lassen. Hier kommt unser optionales Feature die Timeline zum Einsatz, mit der oben genannte Funktion in unser Web-Interface implementiert wird. (siehe 5.3)

6.6 Schnittstelle Datenbank, SNMP-Manager

Da wir unserem SNMP-Manager ermöglichen wollen Daten aus unserer Datenbank auszu-lesen und darin zu speichern, müssen wir diese an ihn anbinden. Dafür bietet sich Java Database Connectivity (JDBC) an, das eine universelle Schnittstelle zwischen Java und einer Vielzahl an Datenbanken bietet.

Die Datenbankanbindung mit JDBC kann in folgende Schritte unterteilt werden:

- Auswahl des JDBC-Treibers
- Verbindung zur Datenbank herstellen
- SQL-Anweisung an die Datenbank schicken
- Aufgebaute Verbindung schließen

Die Programmierung von JDBC ist relativ simpel, da nur wenige Zeilen Programmcode benötigt werden, um eine Datenbankverbindung aufzubauen und einen SQL-Befehl abzusetzen. Der kompliziertere Teil ist die Vorarbeit und besteht darin, einen geeigneten Treiber auszuwählen. Die zentrale Rolle spielen die angesprochenen JDBC-Treiber, die ein Treibermanager verwaltet. Hat man sich für einen Treiber entschieden, muss dieser geladen werden, was in unserem Fall die folgende Code-Zeile bewirkt:

```
Class.forName("com.mysql.jdbc.Driver");
```

Nach dem erfolgreichen Laden des Treibers muss unser Programm eine Verbindung zur Datenbank herstellen. Wir verwenden dafür die Methode `DriverManager.getConnection()`, die Form der Datenbank-URL unter der eine Datenbank zu erreichen ist hängt vom verwendeten Treiber ab. Im Fall von Netview würde der Aufruf folgendermaßen aussehen:

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/netviewdatabase",    User-  
name, Passwort);
```

Nun können beliebige SQL-Statements zu einem String zusammengesetzt werden und an die Datenbank geschickt werden.

Nach jeder abgeschlossenen SQL-Aktion sollte man die Verbindung mit

```
con.close();
```

explizit schließen und so die dafür reservierten Ressourcen freigeben.

7 Marketing

7.1 Logo

Das NetView Logo symbolisiert durch die verbundenen Punkte ein Netzwerk.

Es wurde in Adobe Illustrator umgesetzt. „Adobe Illustrator (kurz Ai) ist ein vektorbasiertes Grafik- und Zeichenprogramm. Es dient also dem Herstellen von Computergrafiken, die man ohne Qualitätsverlust beliebig in ihrer Größe verändern kann – im Unterschied zu pixelbasierten Bildbearbeitungsprogrammen wie z. B. Photoshop.“ [ADI]

Das Logo gibt es in mehreren Varianten:

- Das Originallogo (weißer Hintergrund und transparenter Hintergrund)
- Eine dunkle Version (schwarzer Hintergrund und weiße Schrift)
- Als Coverbild für Facebook und Twitter (ebenfalls dunkle und helle Version)



Abbildung 37: NetView Logo - Transparent

Das Originallogo hat die Abmessungen: 300 x 300 Pixel.

Die Coverbild-Version ist 851 x 315 Pixel groß.

7.2 Schrift

Der Schriftzug „NV – NetView“ hat die Schriftart „Trebuchet MS“, der Untertitel „NetView“ wurde so bearbeitet, dass der Platz zwischen den Buchstaben größer ist und dadurch genau mit den oberen Buchstaben abschließt.

7.3 Hintergrund

Die Punkte im NetView-Logo sind einfache Formen, die teilweise dicke Konturen haben.

Die Farben der Punkte sind mit Adobe Kuler abgestimmt.

Hex-Codes:

- #76BD7E
- #3CAE6A
- #009EE3
- #00A09A
- #A5CEB6

Die Punkte liegen auf der mittleren Ebene, hinter der Schrift und vor den Linien.

Die Linien liegen auf der hintersten Ebene, da es so den Anschein erweckt sie kämen direkt aus den Punkten.

Linien Hex-Code: #D3DDD6

7.4 Social Media

7.4.1 Facebook

Unsere NetView Facebook -Seite wurde zu Projektbeginn erstellt. Sie ist unter der Kategorie Software registriert und zählt 71 Likes.

Das dunkle Logo ist das Profilbild, während, zum Kontrast, das helle Logo das Coverbild ist.

Besonders wichtig ist die Facebook Seite nicht nur wegen dem Bekanntheitsgrad sondern auch da sie gleichzeitig eine Plattform zur Veröffentlichung unseres Videos darstellt.

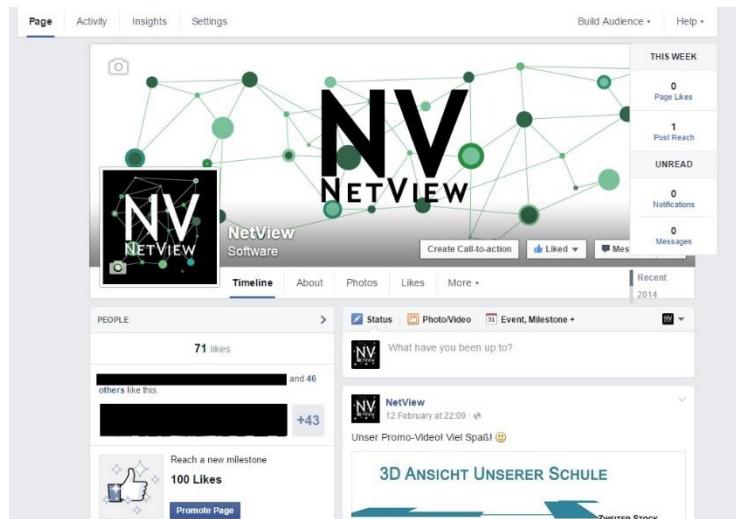


Abbildung 38: facebook-fan-page

7.4.2 Twitter

Die Twitter-Seite wurde gleichzeitig mit der Facebook-Seite aufgesetzt.

Der Unterschied ist, dass das Profilbild auf Twitter das helle Logo ist während das Coverfoto die dunkle Version zeigt.

7.5 Website

Die Website ist mit HTML5 und CSS3 umgesetzt und ist größtenteils an das Template „Twenty“ angelehnt und an unsere Bedürfnisse angepasst.

Mit Hilfe von CSS ist die Website auch responsive, dieses Feature wurde mit Hilfe von mehreren CSS Dateien umgesetzt. Responsive Design bedeutet, dass sich die Website an jede Größe anpasst, also an verschiedene Bildschirmgrößen von PCs, Laptops und mobilen Geräten.

Beispiel:

```
body {  
  min-width: 320px;  
}  
  
h2 {  
  font-size: 1.25em;  
  letter-spacing: 0.1em;  
}  
  
h3 {  
  font-size: 1em;  
  letter-spacing: 0.025em;  
}  
  
p {  
  text-align: justify;  
}
```

Abbildung 39: body aus style-mobile.css

```
header {  
  text-align: center;  
}  
  
header.major {  
  padding-bottom: 0;  
}  
  
header.special {  
  margin-bottom: 3em;  
  padding-left: 1.5em;  
  padding-right: 1.5em;  
}  
  
header.special:before, header.special:after {  
  width: 38%;  
}  
  
header.special .icon {  
  font-size: 0.75em;  
  top: 1.5em;  
}  
  
header p {  
  text-align: center;  
}  
  
footer.major {  
  padding-top: 0;  
}
```

Abbildung 40: header aus style-mobile.css

Insgesamt zählt die NetView Website 9 css Dokumente:

- Fonts-awesome.min.css
- Skel.css
- Style.css
- Style-mobile.css
- Style-narrow.css
- Style-narrower.css
- Style-normal.css
- Style.noscript.css
- Style-wide.css

Die Website ist eine Singel Page Website, das bedeutet, sie besteht nicht aus einzelnen, miteinander verknüpften HTML-Dokumenten, sondern nur aus einem einzigen.

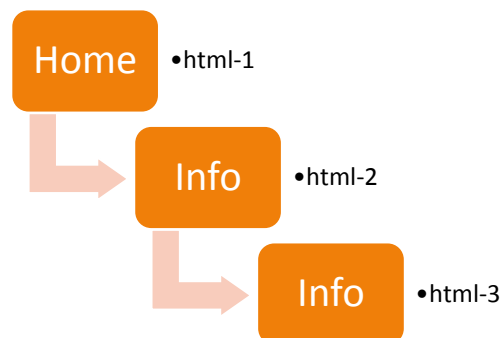


Abbildung 41: "Multipage-Design" bzw. klassische Website

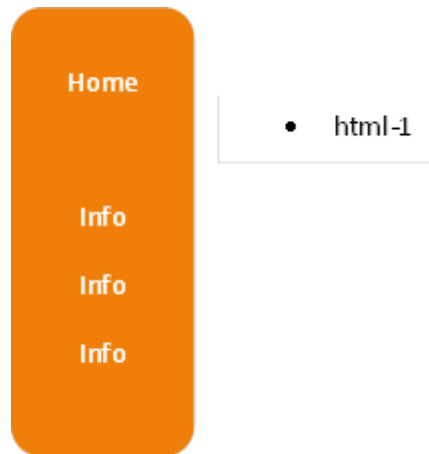


Abbildung 42: "one-page website"

Ein großer Vorteil ist, dass die Website so schneller lädt, weil nur ein Dokument geladen werden muss, nicht mehrere. Allerdings hängt das davon ab wie viel Information in der Seite steckt. Bei zu großen Datenmengen kann es zu längeren Ladezeiten als bei normalen Websites kommen. Weiters kann man sie auch einfacher bearbeiten, da man nur einmal das css oder js Dokument einbinden muss, und Änderungen sofort auf die ganze Website angewendet werden. Außerdem ist diese Art von Website gerade sehr gefragt. Um sie noch ansprechender und moderner zu gestalten, ist die Website in hellen Farben gehalten, dadurch sticht auch das dunkle Logo besonders heraus.

Um die wichtigsten Dinge besonders herauszuheben, sind einige Wörter und Passagen in unserem Text mit dem **Tag** hervorgehoben. Dadurch werden die Wörter fett.

Die Website läuft auf dem Schulserver der HTL Rennweg und hat die Domain `www4.htl.rennweg.at` – das hat den großen Vorteil, dass wir uns bei Problemen direkt an den dafür zuständigen Lehrer wenden können, und außerdem ist die Domain für uns kostenlos.

Die Website ist unterteilt in:

- Welcome
- Projekt
- Downloads
- Team
- Kontakt

Klickt man auf das Icon Welcome findet man eine Abwandlung unseres Logos in Schwarz. Darüber liegt eine semitransparente Box mit einer kurzen Beschreibung von NetView und einem „Mehr dazu“ Button, der direkt zur Projektbeschreibung führt, wenn man ihn anklickt,

Im Punkt "Projekt" ist noch einmal genauer definiert, was NetView kann. Außerdem findet man hier unser Promotionvideo.

Der nächste Punkt „Downloads“ stellt die Dokumente „Antrag“, „Ansuchen“ und „Planung“ zur Verfügung. Die Downloads funktionieren über Dropbox Links.

Als nächstes kommt man zum Icon "Team". Klickt man auf dieses, findet man, zu jedem Teammitglied, Fotos, Namen und kurze Beschreibungen der Aufgaben . (Zuvor hatten wir statt den Fotos Platzhalter-Icons.)

Unter "Kontakt" findet man unsere E-Mail-Adresse, sowie die Schuladresse und zwei Buttons die zu unserem Facebook- und Twitter-Account führen.



Abbildung 43: welcome

7.6 Promotionvideo

7.6.1 Allgemein

Das Promotionvideo für NetView soll Nutzern helfen zu verstehen, was unser Tool ist und wozu man es benutzen kann.

Das Video ist 00:57 Sekunden lang, mit Musik von Bensound hinterlegt, läuft mit 29 fps und ist in MPEG4 gerendert, um die Qualität von Video und Ton zu erhalten.

Das Know-How für die Video Animationen habe ich mir mittels YouTube Tutorials und verschiedene Foren angeeignet.

Die Videoauflösung beträgt 1920x1080, das entspricht einer HD-Auflösung.

Das Video ist in blau und grün gehalten. So passt es gut zu unserem Logo. Das Design ist minimalistisch gehalten.

7.6.2 Inhalt

Das Promotionvideo hilft zu verstehen, was das Tool NetView kann und wozu man es verwenden kann.

Das Intro bildet unser Logo. Es ist so animiert, dass die Punkte sich zuerst von selbst verbinden und dann der NetView Schriftzug erscheint. Die zweite Szene ist der Schriftzug, „Was ist NetView?“, wobei „NetView“ blau auf weißem Hintergrund und „Was ist“ schwarz auf blauem Hintergrund. erscheint. Es folgt die Szene „3D Animation der Schule“, wobei sich die Stockwerke der Schule drehen und dann aufteilen. Die Vorlage dafür war das NetView Tool selbst. Neben den einzelnen Ebenen erscheinen dann die Stockwerknamen. Dann dreht sich alles wieder zurück und die unteren Stockwerke verschwinden. Übrig bleibt der 2. Stock. Hier wird herangezoomt und durch eine kleine Animation verdeutlicht, dass man zum Beispiel die Auslastung zwischen Switches und Access-Points auslesen kann. Danach kommt wieder ein Schriftzug, mit demselben Format wie „Was ist NetView?“. Dieser Schriftzug lautet: „Was kann NetView?“. Danach kommt eine Textszene mit einer kleinen Animation, die ein W-LAN Zeichen eines PCs darstellt. Der Text lautet: „NetView zeigt die Auslastung auf einer beliebigen Leitung“. Als nächstes kommt eine Animation, die einen Rechner zeigt, auf dem die Auslastung einer Leitung angezeigt wird, darüber steht „NetView“. Es folgt eine weitere Textanimation. Der Text kommt von oben und unten und bildet den Satz: „NetView gibt einen Überblick über das Schulnetz und hilft so bei der Fehlerbehebung“. Dann ist ein Ladebalken zu sehen. Dieser „lädt“ die nächste Szene „Sekündliche Updates“. Zusehen ist eine Uhr, deren Sekundenzeiger sich bewegt. Die vorletzte Szene zeigt den Link zu unserer Website, sowie die E-Mail-Adresse und die Namen unserer Social-Media-Seiten. Zuletzt ist der Link zur Website "bensound.com" zu sehen, da von dort die Musik für unser Video stammt. Der Musiktitel selbst heißt „Moose“.

7.6.3 Videoerstellung

7.6.3.1 Musik

Musik ist ein wichtiger Faktor bei Videos, da sie verschiedene Stimmungen hervorrufen kann. Videos ohne Musik werden häufig als langweilig empfunden. Daher war von Anfang an klar, dass das Video mit Musik hinterlegt sein wird.

Wichtig für die Musikfindung war als Erstes eine Seite zu finden, auf der Gratismusik angeboten wird. Hierbei bin ich auf Bensound gestoßen:

“Welcome to Bensound.com, here you can download my royalty free music at no charge for Youtube and your multimedia projects. “[BEN]

Auf dieser Seite wurden folgende Musikstücke ausgewählt und ausprobiert:

- Deepblue.mp3
- Epic.mp3
- Funkyelement.mp3
- Moose.mp3
- Newdawn.mp3
- Psychedelic.mp3
- Sci-fi.mp3
- Straight.mp3

Endgültig in unser Promotion-Video eingebaut ist nun “moose.mp3”, da dieses die beste Stimmung erzeugt.

7.6.3.1.1 Lizenz

Die Musik auf bensound und somit auch unser Video betreffend ist eine sogenannte „creative common license“. Die Kurzfassung, welche auf der bensound Website veröffentlicht ist, lautet:

“You can use for free my music licensed under the Creative Commons License in your multimedia project (online videos, websites, animations, etc.) as long as you credit me with a

link to my Website. Examples of proper way to credit me: "Music:<http://www.bensound.com/royalty-free-music> or "Music: Song title - Bensound.com"" [LIZ]

Diese Lizenz erfüllen wir mit der EndEinstellung unseres Videos, welche lautet:

„Music by: <http://www.bensound.com>“

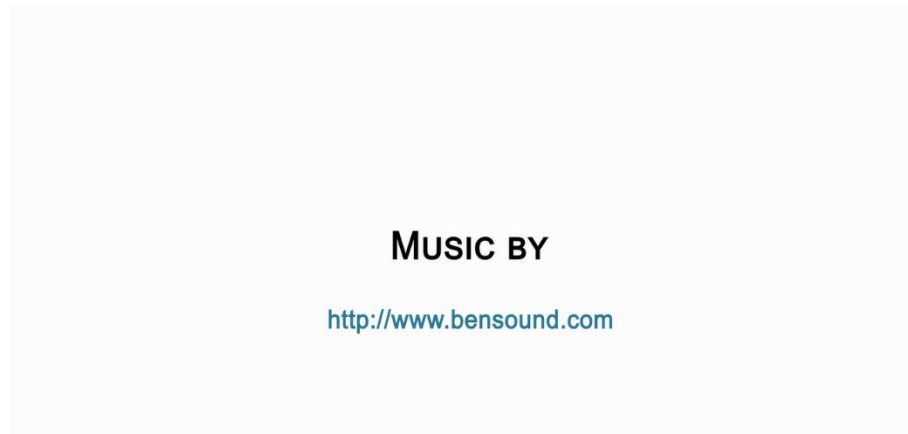


Abbildung 44: "Music by bensound"

7.6.4 Szenen

7.6.4.1 Intro

Der erste Entwurf des Intros hat vorgesehen, dass sich das „Netzwerk“, also die mit Linien verbundenen Punkte, dreidimensional um sich selbst drehen. Um diesen Effekt zu erzielen, habe ich zuerst mit Hilfe eines Scripts mehrere Würfel erzeugt, diese dann auf eine 3D-Ebene gelegt und sie in der Tiefe verschoben, um den 3D Effekt zu verdeutlichen. Die Würfel waren dabei Platzhalter für die Kugeln die später eingebaut werden sollten.

Das Ganze habe ich dann mit Linien verbunden und mit Hilfe eines Null-Objekts um sich selbst gedreht.

Später hat sich herausgestellt, dass dieser Effekt mit Kugeln nicht funktioniert, da man diese zwar durch Schatten dreidimensional darstellen kann, sie allerdings zweidimensional sind, wenn man sie um sich selbst dreht.

Um dieses Problem zu umgehen, habe ich die andere Variante entwickelt, die sich verbindenden Punkte, die dann das Logo bilden.

Die Formen liegen alle auf demselben Shape-Layer, also Formebenen für Vektorgrafiken. Eine Form besteht aus einem Pfad, einer Kontur und einer Fläche. Allerdings kann man die Fläche und/oder die Kontur ausblenden.

Die Formen werden durch Pfade verbunden, die auf einer Ebene liegen, die unter dem oben erwähnten Shape-Layer liegt. Damit wird erreicht, dass es aussieht als würden die Linien direkt aus den Formen kommen.

Beim Verbinden der Pfade gab es das Problem, dass diese sich, wenn man mehrere Pfade auf einmal transformieren will, nicht mehr ausdehnen, sondern aufklappen und dann ausdehnen. Die Lösung war, jeden Pfad einzeln zu Animieren.

Die am Ende des Intros eingeblendete Schrift ist Arial, mit der Einstellung Narrow.

Die Buchstaben NV und der Schriftzug NetView liegen auf einem Text-Layer. Diese Ebene hat die opacity 0, dass bewirkt, dass die Schrift nicht sichtbar ist. Die Schrift wird erst dann sichtbar, wenn ein Keyframe die opacity hoch stellt.

Der NetView Schriftzug hat einen Aufklappeffekt, der mittels Modifizieren des x-Wertes erzielt wurde. Dieser Wert ist standartmäßig mit dem y-Wert verkettet. Um den Aufklappeffekt zu erzielen muss die Verkettung erst aufgehoben werden. Dann wird der x-Wert auf 0 gesetzt und in der Zeitleiste an dem Punkt des Effektbeginns ein Keyframe gesetzt. Der zweite Keyframe wird gesetzt, wenn die Animation zu Ende sein soll, also die Schrift ihre volle Größe erreicht haben soll. An diesem Zeitpunkt wird der X-Wert dann auf 100 gestellt.

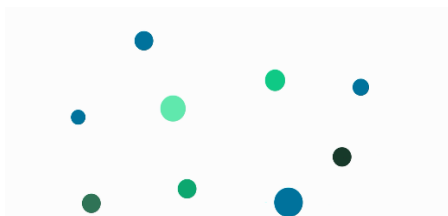


Abbildung 45: Intro1

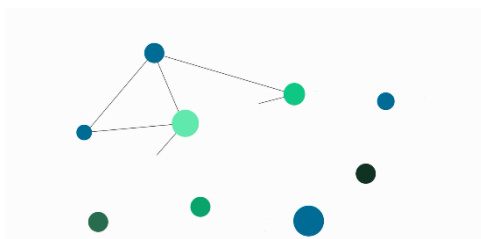


Abbildung 46: Intro2



Abbildung 47: Intro3

Von den einzelnen Punkten gehen Linien aus. Diese verbinden alle Punkte und bilden dann ein „Netzwerk“

Nachdem alle Punkte verbunden sind wird zuerst die Abkürzung „NV“ und dann der ganze Projektname „NetView“ eingeblendet.

7.6.4.2 3D-Animation der Schule

Die Vorlage für diese Szene war das eigentliche Tool NetView. Am Beginn sieht man die drauf-sicht, dann dreht sich die Ansicht und die einzelnen Ebenen die die Stockwerke bilden Klappen auf.

Diese Drehung wurde erreicht, indem in der Mitte der Animation ein Null-Objekt Platziert wurde. Nullobjekte sind Transparente Objekte, die mit allen Ebenen verknüpft werden können und helfen so eine Animation die auf mehreren Ebenen aufgeführt werden soll, synchron auszuführen. Die Ebenen werden mit dem Objekt verknüpft und die Animation wird nur im Nullobjekt eingestellt.

Es ist auch möglich eine Nullobjekt-Animation auszuführen, und gleichzeitig eine Animation in der Ebene selbst auszuführen.

Die verschiedenen Ebenen sind abwechselnd in einer helleren und dunkleren Farbe gewählt um das ganze Modell übersichtlicher zu gestalten.

Nachdem der Text zu den jeweiligen Stockwerken eingeblendet ist, verschwinden alle Stockwerke, bis auf den 2. Stock, langsam. Dieser Effekt wurde erzielt indem die Opacity von 100 langsam mit Keyframes auf 0 gestellt wird.

Die Rückdrehung entsteht wieder durch das Null-Objekt.

Die Animation in zwischen Switch und Access-Point ist dadurch erreicht, dass nicht die Kamera, sondern das ganze Bild mit Hilfe von Pfaden verschoben wird.

Die % Zahlen sind mit der „Numbers“ Funktion umgesetzt. Diese Funktion erlaubt es, eine bestimmte Nummer(Zahl1) einzustellen, mit oder ohne Dezimalstelle, und mit Hilfe von Keyframes an einer anderen Stelle eine andere Zahl (Zahl2(einzustellen. Wenn man den Clip nun abspielt zählt die Funktion zu von Zahl1 zu Zahl2 hinauf oder hinunter.

Auf diese Weise sind auch die PC-Animation und die Laden-Animation entstanden.



Abbildung 48: 3D Ansicht der Schule 1



3D ANSICHT UNSERER SCHULE

Abbildung 49: 3D Ansicht der Schule 2



3D ANSICHT UNSERER SCHULE

Abbildung 51: 3D Ansicht der Schule 3



3D ANSICHT UNSERER SCHULE

Abbildung 50: 3D Ansicht der Schule 4

Die Etagen werden von oben gezeigt. Dann wird die Sicht gedreht und der Text „3D Ansicht Unserer Schule“ angezeigt.

Die Etagen Teilen sich auf, jede zweite hat eine hellere Farbe.

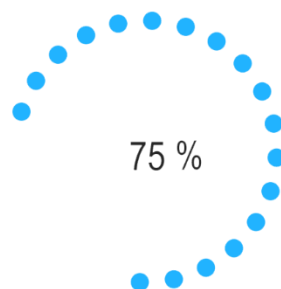
Nachdem alle Etagen voll ausgefahren sind werden die Etagen-namen eingeblendet.

7.6.4.3 Ladekreis

Um das Laden einer Sequenz zu symbolisieren, wurde eine Szene entwickelt auf der ein Lade-Kreis zu sehen ist. In dem Kreis befinden sich zahlen.

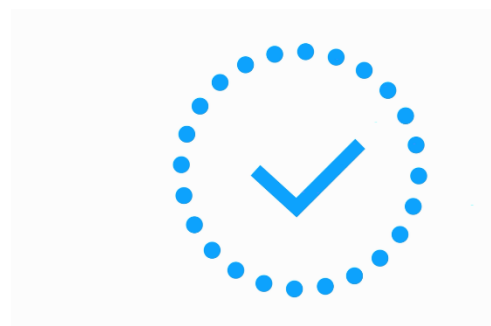
Um den lade-effekt zu erzielen erstellt man als erstes einen Kreis. Wenn man nun auf in die Ebene dieses Kreises geht, findet man den Ordner „Contents“ und daneben Add. In dem Add Menü befindet sich der Effekt „Repeater“, dieser wird ausgewählt und die Anzahl der gewünschten Kopien werden definiert. Im Repeater Effekt gibt es den Ordner: Transform Repeater, hier muss die Position auf 0 gesetzt werden. So liegen alle Kopien übereinander. Nun kann man die rotation-Werte so modifizieren, dass die Kopien einen Kreis ergeben. Um die Animation zu erzielen wird nicht die Rotation, sondern es werden die Kopien animiert, d.h. am Beginn der Animation werden die Kopien auf 0 oder 1 gesetzt und ein Keyframe gesetzt, am Ende setzt man die Kopien so, dass sie wieder einen Kreis bilden, hier wird wieder ein Keyframe gesetzt. Nun ist der Kreis animiert.

Für die % Angabe in der Mitte des Kreises wurde wieder die Numbers Funktion (siehe „3D animation der Schule“) angewendet.



Gleichzeitig mit den Zahlen in der Mitte des Kreises werden auch die Punkte verändert.

Abbildung 52: "Lade-Kreis"



Am Ende ist der Kreis geschlossen und ein „Häckchen-Symbol“ wird statt den Zahlen eingeblendet.

Abbildung 53: Lade-Kreis geladen

7.6.4.4 Sekündliche Updates

Für diese Szene wurde eine Uhr direkt in After Effects gezeichnet, diese bleibt die ganze Zeit statisch, der Sekundenzeiger liegt auf einer höheren Ebene, er besteht aus einem einzelnen Pfad, dessen Position mit Hilfe von Keyframes verändert und animiert wird.

Wichtig ist es dabei nur einen, den äußeren, Ankerpunkt des Pfades zu verschieben, und den Pfad immer gleich lang zu halten, da sich sonst der ganze Pfad, und nicht nur dessen Position verändert.



Abbildung 54: Sekündliche Updates

7.6.5 Zusammenfügen

Zum Zusammenfügen der einzelnen erstellten Kompositionen wurden pro Komposition neue Spuren benützt.

Da ich die Szenen einzeln animiert habe, ist jeder Teil eine eigene Komposition, das ist einerseits hilfreich für das zusammenfügen der einzelnen Kompositionen, andererseits ist es auch praktisch für das nachträgliche bearbeiten, falls eine Szene doch an einen anderen Platz gehört bzw. falls die Länge einer einzelnen Szene verändert werden muss. Außerdem ermöglichen die vielen einzelnen Sequenzen eine leichtere Fehlerbehebung, sollte eine Szene aus irgendeinem Grund verloren gehen oder fehlerhaft sein.

Für die Musik gibt es eine eigene Audiospur, die das ganze Video hindurchläuft und am Ende mit Hilfe von Keyframes leiser wird. Dazu muss man einen Keyframe an dem Zeitpunkt setzen, an dem die Musik beginnen soll leiser zu werden, und einen an dem sie ihren leisesten Punkt erreichen soll. Dann stellt man bei dem zuletzt genannten Keyframe die gewünschte dB-Rate ein.

7.6.6 Rendering

Als Rendern bezeichnet man das Erstellen der Frames eines Films aus einer Komposition. Beim Rendern eines Frames wird ein kombiniertes, zweidimensionales Bild aus allen Ebenen, Einstellungen und sonstigen Informationen in einer Komposition erstellt, aus denen

das Modell für dieses Bild besteht. Beim Rendern eines Films werden alle Frames, aus denen der Film besteht, jeweils einzeln gerendert.

Obwohl sich der Begriff Rendern im normalen Sprachgebrauch auf die Endausgabe bezieht, sind die Vorgänge beim Erstellen von Ansichten zur Vorschau im Footage-, Ebenen- und Kompositionsfenster ebenfalls eine Art Rendern, denn eine RAM-Vorschau lässt sich als Film speichern und als Endausgabe verwenden.

Nach dem Rendern für die endgültige Ausgabe wird die Komposition von einem oder mehreren Ausgabemodulen verarbeitet, mit dem bzw. denen die gerenderten Frames in einer oder mehreren Ausgabedateien kodiert werden. Dieser Prozess der Kodierung gerendeter Frames in Dateien für die Ausgabe stellt eine Art des Exportierens dar. [REN]

Gerendert und exportiert wurde in mp4 da MPEG4 viele Vorteile gegenüber anderen codecs aufweist:

- offizieller Container des mpeg-4 Standards
- wurde speziell für mpeg-4 Audio/Video entwickelt, das heißt man hat um Beispiel extrem niedrige Overheads
- aus demselben Grund hat mp4 auch nicht Probleme mit mpeg-4 wie .avi:
- da keine fourcs wie in .avi verwendet werden braucht man nur einen einzigen Decoder für alle mp4 Files (egal ob mit divx5, xvid, 3ivx oder was auch immer erstellt wurde)
- keinerlei Probleme mit vbr mp3, was nur mit einem hack in .avi möglich ist
- keinerlei Probleme mit b-frames, die ebenfalls in .avi nur mit hack möglich sind
- mit AAC unterstützt mp4 auch Multichannel auf niedrigen Bitrate (~192kbps) während ac3 300-400kbps dafür benötigt
- dazu unterstützt mp4 auch Dinge wie seeking und cutten nicht nur bei keyframes (mp4 wurde speziell mit dem Ziel entwickelt gut bearbeitbar zu sein)
- weiters kann mp4 alle gängigen a/v Formate Händeln, die es gibt im mpeg-4 Standard sind folgende speziell angeführt/spezifiziert:
- Audio: mp3 und AAC

- Video: mpeg-4 part2 (kennen wir als divx5, xvid...) und avc (h.264), quasi der Gegenpart von AAC
in einer gewissen Weise ist es natürlich sinnvoll den Umfang der möglichen codecs einzuschränken, da man nicht erwarten kann, dass z.B.: ein Hardware Player alle Formate unterstützen kann.
Dennoch erlaubt der mpeg-4 Standard im mp4 Container andere codecs abzuspeichern, z.B.: ist es bereits möglich vorbif und vorsubs auf diese Weise zu verwenden
- sogenannte "user interactivity" (bifs)
dieses System erlaubt es interaktive mp4 Dateien zu erstellen, z.B.: ist es damit möglich spiele (wie bei Flash) zu erstellen, Streaming Menüs oder Menüs wie auf DVDs“
[MPG]

Literaturverzeichnis

[MIT] "The MIT License (MIT)". URL: opensource.org/licenses/MIT (Stand 04.01.2015)

[CIU] "Can I Use... Support Tables for HTML5, CSS3, etc.". URL: caniuse.com/webgl (Stand 04.01.2015)

[BAW] "WebGL Public Wiki". URL: khronos.org/webgl/wiki/BlacklistsAndWhitelists (Stand 04.01.2015)

[TJS] "mrdoob/three.js – GitHub"

- theo-armour (06.05.2012). "Features" (Aktualisiert: 26.11.2012). URL: github.com/mrdoob/three.js/wiki/Features (Stand 04.01.2015)
- mrdoob (22.04.2012). "Readme" (Aktualisiert: 11.09.2014). URL: github.com/mrdoob/three.js/blob/master/README.md (Stand 04.01.2015)
- "Releases". URL: github.com/mrdoob/three.js/releases (Stand 04.01.2015)

[AIL] URL: <http://www.eemedia.at/tag/illustrator/>

[AEF] URL: <http://adobe-after-effects.giga.de>

[SNM] URL: <http://tools.ietf.org/html/rfc1157>

[DUD] URL: <http://www.duden.de/rechtschreibung/Bandbreite>

[CI1] URL: <http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en&translate=Translate&objectInput=1.3.6.1.2.1.2.2.1.10>

[CI2] URL: <http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en&translate=Translate&objectInput=1.3.6.1.2.1.2.2.1.16>

[CI3] URL: <http://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/8141-calculate-bandwidth-snmp.html>

[IET] URL: <https://tools.ietf.org/html/rfc3417>

[WIK] URL: <http://de.wikipedia.org/wiki/Session>

[OID] URL: http://www.networkmanagementsoftware.com/wp-content/uploads/SNMP_OID_MIB_Tree.png

[BEN] URL: <http://www.bensound.com/> (Stand 07.01.2015)

[LIZ] URL: <http://www.bensound.com/licensing> (Stand 07.01.2015)

[REN] URL: <https://helpx.adobe.com/de/after-effects/using/basics-rendering-exporting.html> (Stand 07.01.2015)

[MPG] URL: <http://forum.gleitz.info/archive/index.php/t-9873.html> (Stand 07.01.2015)

[UNIX] URL: <http://www.unixtimestamp.de/> (Stand 27.3.2015)

[SQL] URL: <http://nilsinfo.de/details/mysqlvorteile.html> (Aktualisiert: 26.01.2004) (Stand 27.3.2015)

Tabellenverzeichnis

Tabelle 1: Im Projektantrag definierte Meilensteine	14
Tabelle 2: SNMPv3-Security Level	21
Tabelle 3: Primarykeys	46
Tabelle 4: NULL-Werte	48

Abbildungsverzeichnis

Abbildung 1: phpMyAdmin.....	16
Abbildung 2: XAMPP Control Panel.....	17
Abbildung 3: UML Klassendiagramm	24
Abbildung 4: LoadJDBCdriver-Methode	25
Abbildung 5: CreateSnmp-Methode.....	25
Abbildung 6: Process-Methode	26
Abbildung 7: ReadAgents-Methode.....	26
Abbildung 8: FetchAgents-Methode.....	27
Abbildung 9: FetchPorts-Methode.....	27
Abbildung 10: Attribute der Agent-Klasse	28
Abbildung 11: SendGet-Methode	28
Abbildung 12: OnResponse-Methode 1	29
Abbildung 13: OnResponse-Methode 2	30
Abbildung 14: HandleResponsePDU-Methode	31
Abbildung 15: CreateNetViewPDU-Methode	31
Abbildung 16: OID-Baum [OID]	32
Abbildung 17.....	32
Abbildung 18: Funktion zum Erstellen der Leinwand	33
Abbildung 19: Erstellen einer perspektivischen Kamera	33
Abbildung 20: Blickfeld einer perspektivischen Kamera.....	34
Abbildung 21: Erstellen einer orthographischen Kamera.....	34
Abbildung 22: Blickfeld einer orthographischen Kamera	35
Abbildung 23: Digitalisierte Version der Pläne mit Raumnummern	36
Abbildung 24: Raum Plan 3. Stock (Foto des Originals)	36
Abbildung 25: Leeres Gebäudemodell im 3D-Interface.....	37
Abbildung 26: Entstehung des Router-Modells	38
Abbildung 27: Zwei der fertigen Modelle	38
Abbildung 28: Beispiel eines Datensatzes im JSON-Format	39
Abbildung 29: PHP-File mit Datenbankabfrage	40
Abbildung 30: Aufruf des PHP-Files beim Laden der Seite	41
Abbildung 31: Farbtabelle.....	42
Abbildung 32: Datumswahl und Schieberegler der Timeline	44
Abbildung 33: Erstellen des Sliders	44
Abbildung 34: ER-Modell.....	46
Abbildung 35: Beispiel dev-Tabelle	50
Abbildung 36: Beispiel if-Tabelle	50
Abbildung 37: NetView Logo - Transparent.....	53
Abbildung 38: facebook-fan-page	55
Abbildung 39: body aus style-mobile.css.....	56
Abbildung 40: header aus style-mobile.css.....	56
Abbildung 41: "Multipage-Design" bzw. klassische Website.....	57
Abbildung 42: "one-page website"	58
Abbildung 43: welcome.....	59
Abbildung 44: "Music by bensound"	62

Abbildung 45: Intro1	63
Abbildung 46: Intro2	63
Abbildung 47: Intro3	63
Abbildung 49: 3D Ansicht der Schule 1	65
Abbildung 50: 3D Ansicht der Schule 2	65
Abbildung 51: 3D Ansicht der Schule 4	65
Abbildung 52: 3D Ansicht der Schule 3	65
Abbildung 53: "Lade-Kreis"	66
Abbildung 54: Lade-Kreis geladen.....	66
Abbildung 55: Sekündliche Updates	67

Autorenverzeichnis

Sabine Schimpf	<ul style="list-style-type: none"> 1 Einleitung 1.1 Projektidee 1.2 Projektteam 1.2.1 Sabine Schimpf 2 Projektkultur 2.1 Dokumentenablage 2.2 Kommunikation 2.2.1 Kurze Besprechungen 2.2.2 Jour fixe 2.2.3 Meetings 2.2.4 Mailverkehr 3 Planung 3.1 Projektziele 3.1.1 Muss-Ziele 3.1.2 Kann-Ziele 3.1.3 Nicht-Ziele 3.2 Projektmanagement 3.2.1 Zeitplanung 3.3 Meilensteine 4 Verwendete Technologien 4.1 Three.js 4.1.1 Browser-Support 4.2 XAMPP 4.7 Hosting 5 Programmierung 5.2 3D-Umgebung, Web-Interface 5.2.1 Szenenaufbau in Three.js 5.2.2 Kamera 5.2.3 Modelle 5.2.4 Antialiasing 5.2.5 Werte 5.2.6 Aufbau des dreidimensionalen Raums 5.2.7 Übersichtlichkeit 5.3 Timeline
Abigail Steinhardt	<ul style="list-style-type: none"> 1.2.4 Abigail Steinhardt 4.3 Adobe Illustrator 4.4 Adobe After Effects 4.5 HTML5 4.5.1 Neuerungen 4.5.2 Vorteile von HTML5 4.6 CSS3

	4.6.1 Vorteile 4.6.2 Nachteile 7 Marketing 7.1 Logo 7.2 Schrift 7.3 Hintergrund 7.4 Social Media 7.4.1 Facebook 7.4.2 Twitter 7.5 Website 7.6 Promotionvideo 7.6.1 Allgemein 7.6.2 Inhalt 7.6.3 Videoerstellung 7.6.4 Szenen 7.6.5 Zusammenfügen 7.6.6 Rendering
Gregor Ivancsics	Kurzfassung Abstract 1.2.3 Gregor Ivancsics 5 Programmierung 5.1 Der SNMP-Manager 5.1.1 Einführung 5.1.2 Bandbreitenmessung und Auslastung 5.1.3 Aufbau
Martin Hammerbacher	1.2.2 Martin Hammerbacher 4.8 SNMP 4.8.1 Funktionsweise 4.8.2 Management Information Base 4.8.3 SNMP Versionen 6 Datenbank 6.1 Anforderungen 6.1.1 Wahl der Datenbanksoftware 6.2 Modell 6.3 Datenstruktur 6.4 Datenbankeinträge 6.5 Timestamp 6.6 Schnittstelle Datenbank, SNMP-Manager