

Berufliches Schulzentrum für Elektrotechnik Dresden

Fachbereich Informationstechnik

Projektdokumentation

Lernfeld 9 - Projekt 3

Auftraggeber: Doubtful-Joy SE

Auftragnehmer: High-Secure GmbH - Projektteam IT20/2 Gruppe 7

Auftragsdatum: 2021.11.15

Historie:

Inhaltsverzeichnis

Abkürzungsverzeichnis

1	Pflichtenheft	1
1.1	Auftraggeber und Auftragnehmer	1
1.2	Ausgangslage	1
1.3	Projektziel	1
1.4	Funktionsspezifikation	2
1.5	Datenspezifikation	2
1.6	Schnittstellenspezifikation	4
1.7	Rahmenbedingungen	4
1.8	Qualitätsbetrachtung	5
1.9	Projektplanung	5
1.10	Kosten-Nutzen-Analyse	6
2	Auswertung und Reflexion	7
2.1	Ablaufdokumentation	7
2.2	Einrichtung IPFire	7
2.3	Einrichtung Admin-PC	8
2.4	Einrichtung DHCP-DNS-DB-Server	9
2.4.1	Einrichtung der CentOS-Installation	9
2.4.2	Einrichtung des DHCP und DNS	10
2.4.3	Einrichtung des Datenbank-Servers	11
2.5	Einrichtung Webserver	13
2.5.1	Einrichtung der CentOS-Installation	13
2.5.2	Einrichtung des Backends	13
2.5.3	Einrichtung des Frontends	13
2.6	Soll-Ist-Vergleich	13
2.7	Abweichung zum Zeitplanung	13
2.8	Optimierungsvorschläge zur Projektrealisierung	14
	Abbildungsverzeichnis	15
	Tabellenverzeichnis	16
	Listings	17
	Literaturverzeichnis	18

Abkürzungsverzeichnis

API	Application Programming Interface	13
CRUD	create, read, update und delete	13
DB	Datenbank	8
DHCP	Dynamic Host Configuration Protocol	1
DNS	Domain Name System	1
DMZ	Demilitarisierte Zone	3
SSH	Secure Shell	8
VM	virtuelle Machine	7

1 Pflichtenheft

1.1 Auftraggeber und Auftragnehmer

Beim Auftraggeber handelt es sich um die Gaming-Plattform **Doubtful-Joy SE**. Ansprechpartner sind

Tabelle 1.1: Ansprechpartner Auftraggeber

Funktion	Name	Vorname	Email
Auftraggeber	Hempel	Steffen	⟨hempel@bszetdd.lernsax.de⟩

Beim Auftragnehmer handelt es sich um das **High-Secure GmbH - Projektteam IT20/2 Gruppe 7**. Ansprechpartner sind

Tabelle 1.2: Ansprechpartner Auftragnehmer

Funktion	Name	Vorname	Email
Projektmanager	Egermann	Péter	⟨i20egermannpe@bszetdd.lernsax.de⟩
Teamleiter	Leyrer	Johannes	⟨i20leyrerjo@bszetdd.lernsax.de⟩
Netzwerkingenieur	Brethfeld	Vinzenz	⟨i20brethfeldvi@bszetdd.lernsax.de⟩

1.2 Ausgangslage

Die existierende Support-Infrastruktur der Gaming-Plattform Doubtful-Joy SE lässt sich über Mail und Telefon kontaktieren. Dabei wird jeder Anruf und jede Mail individuell von einem Mitarbeiter als Ticket gespeichert und in einem zentralen Laufwerk abgelegt. Effizienz, Ordnung und Übersichtlichkeit sind nicht ausreichend vorhanden.

1.3 Projektziel

Die Gaming-Plattform Doubtful-Joy SE möchte ihre existierende Support-Infrastruktur durch ein Ticketsystem ersetzen. Dieses soll für Kunden und Mitarbeiter über ein Web-Interface erreichbar sein. Tickets sollen über dieses direkt erstellt und mit beliebig vielen Attachments versehen werden können.

Außerdem soll eine Segmentierung der Netzinfrastruktur mit einer sichereren Trennung von öffentlich erreichbaren Diensten und dem Intranet eingerichtet werden. Ebenso sollen die internen Dienste Domain Name System (DNS) und Dynamic Host Configuration Protocol (DHCP) auf einem separatem System bereitgestellt werden, um eine Abhängigkeit von der Firewall auszu-schließen.

Doubtful-Joy SE setzt auf RedHat und binärkompatible Systeme, weshalb diese System-Strategie weiterhin umgesetzt werden soll.

1.4 Funktionsspezifikation

Von der Realisierung sind betroffen:

Manware

- Projektteam IT20/2 Gruppe 7
- Support-Mitarbeiter des Auftraggeber
- IT-Mitarbeiter des Auftraggebers

Orgware

- Sicherheitsanforderungen
- Benutzerhandbuch
- Benutzerschulung

Hardware

- Server
- Mitarbeiter-PCs

Software

- VM-Ware
- Datenbank-Server
- Web-Server
- Firewall-System
- DNS
- DHCP

1.5 Datenspezifikation

Da von etwa 1000 Telefonanrufen und Emails pro Tag ausgegangen wird, kann dies etwa 1:1 in 1000 Tickets übertragen werden. Der Speicherbedarf pro Ticket wird hier im Schnitt auf etwa 5 MB geschätzt, da wahrscheinlich häufiger Anhänge in Bildform zur besseren Problembeschreibung genutzt werden. Zusätzlich wird davon ausgegangen, dass die Daten zur Sicherheit und Nachvollziehbarkeit für ein Jahr gespeichert werden, wodurch die Datenbank 1830 GB Speicher in einem Jahr benötigt.

$$\frac{5 \text{ MB}}{\text{Ticket}} \cdot \frac{1000 \text{ Ticket}}{\text{Tag}} = \frac{5000 \text{ MB}}{\text{Tag}}$$

$$\frac{5000 \text{ MB}}{\text{Tag}} \cdot 365 \text{ Tage} = \frac{1\,825\,000 \text{ MB}}{\text{Jahr}} \stackrel{\wedge}{=} \frac{1830 \text{ GB}}{\text{Jahr}}$$

Da es keine Good-Practice ist, die Bilder in der Datenbank zu speichern, wird nur der Dateipfad zu den Bildern in der Datenbank hinterlegt, die Bilder selbst liegen auf der Festplatte des Webservers. Damit verringert sich der geschätzte Speicherbedarf der Datenbank auf etwa 183 GB pro Jahr.

$$\frac{0,5 \text{ MB}}{\text{Ticket}} \cdot \frac{\text{Ticket}}{\text{Tag}} = \frac{500 \text{ MB}}{\text{Tag}} \stackrel{\wedge}{=} \frac{183 \text{ GB}}{\text{Jahr}}$$

Die Bilder selbst benötigen zum aktuellen Stand auf der Festplatte 1643 GB Speicher pro Jahr.

$$\frac{4,5 \text{ MB}}{\text{Bild}} \cdot 1000 \frac{\text{Bild}}{\text{Tag}} = \frac{4500 \text{ MB}}{\text{Tag}} \stackrel{\wedge}{=} \frac{1643 \text{ GB}}{\text{Jahr}}$$

Die Art von Daten sind personenbezogene Daten in Text- und Bildform.

Der Datenfluss geht vom Clienten zur Demilitarisierte Zone (DMZ) und zur Bearbeitung dann zum PC des Support-Mitarbeiters, grafisch dargestellt in Abb. 1.1 auf der nächsten Seite.

1.6 Schnittstellenspezifikation

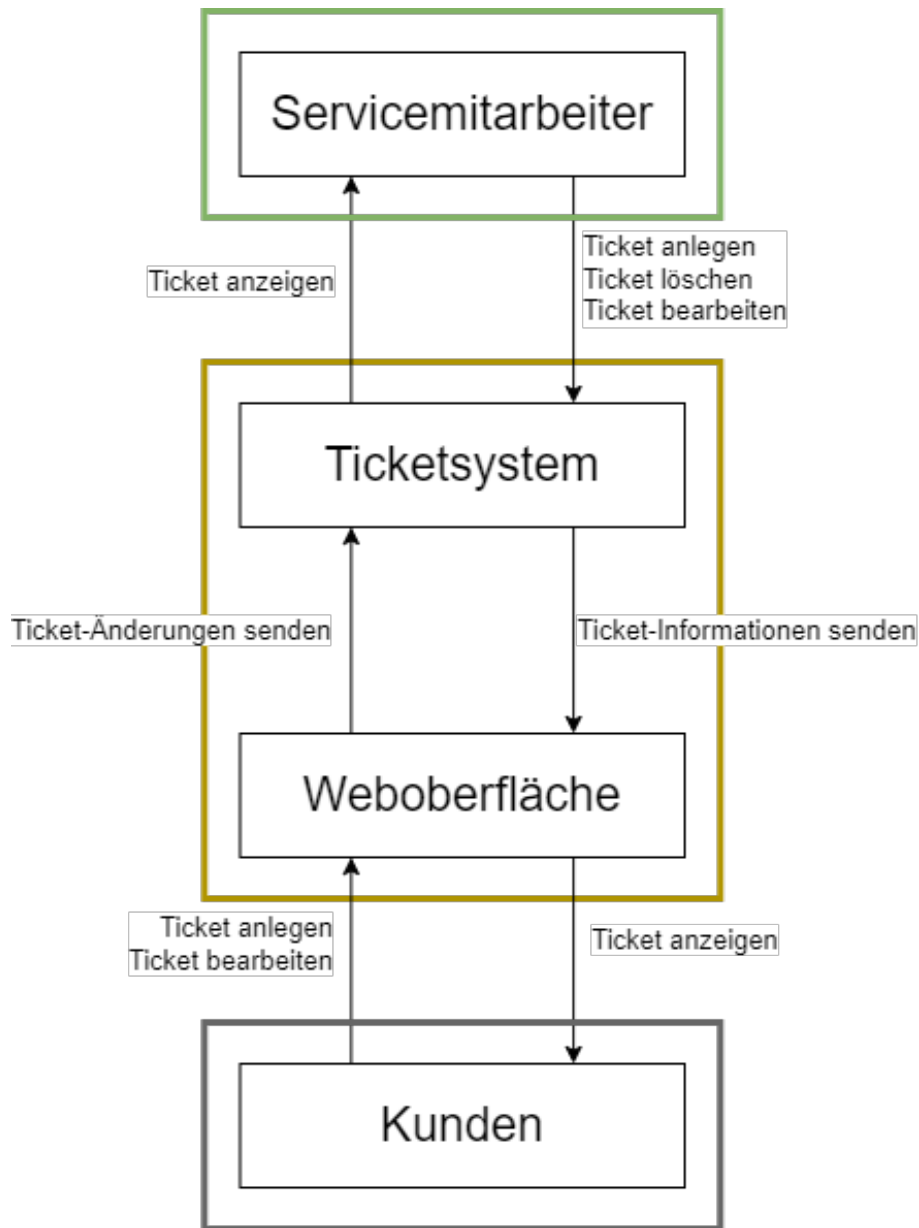


Abbildung 1.1: Schnittstellenspezifikation

1.7 Rahmenbedingungen

Der Auftraggeber hat folgende Ressourcen bereitzustellen und Mitwirkungspflichten:

- Server
- Mitarbeiter-PCs
- Zugriff auf alle zu bearbeitenden Systeme und Zutritt zu den notwendigen Räumlichkeiten
- Kooperation und eventuell notwendigen lokalen Support

1.8 Qualitätsbetrachtung

Die Arbeitspakete werden stets während der Bearbeitung sowie nach der Fertigstellung auf Funktion und Qualität überprüft.

Wöchentlich werden Meetings abgehalten um den Stand des Projekts zu erörtern und auf eventuell auftretende Probleme zeitnah reagieren zu können.

Die Zeitplanung und damit der Aufwand ist in Abb. 1.3 auf der nächsten Seite in kleinem Format und groß in Abb. A.1 auf Seite 20 zu sehen. Für einen langfristigen Support für nach der der Fertigstellung wird ein zusätzliches Angebot vorgelegt.

1.9 Projektplanung

Die Projektplanung ist im Projektstrukturplan, zu sehen in Abb. 1.2, und im Gantt-Diagramm, zu sehen in Abb. 1.3 auf der nächsten Seite, bzw. Abb. A.1 auf Seite 20, abgebildet. Ebenso wird der im Anhang Seite 22 zu betrachtende Netzwerkplan Abb. B.1 auf Seite 22 umgesetzt.

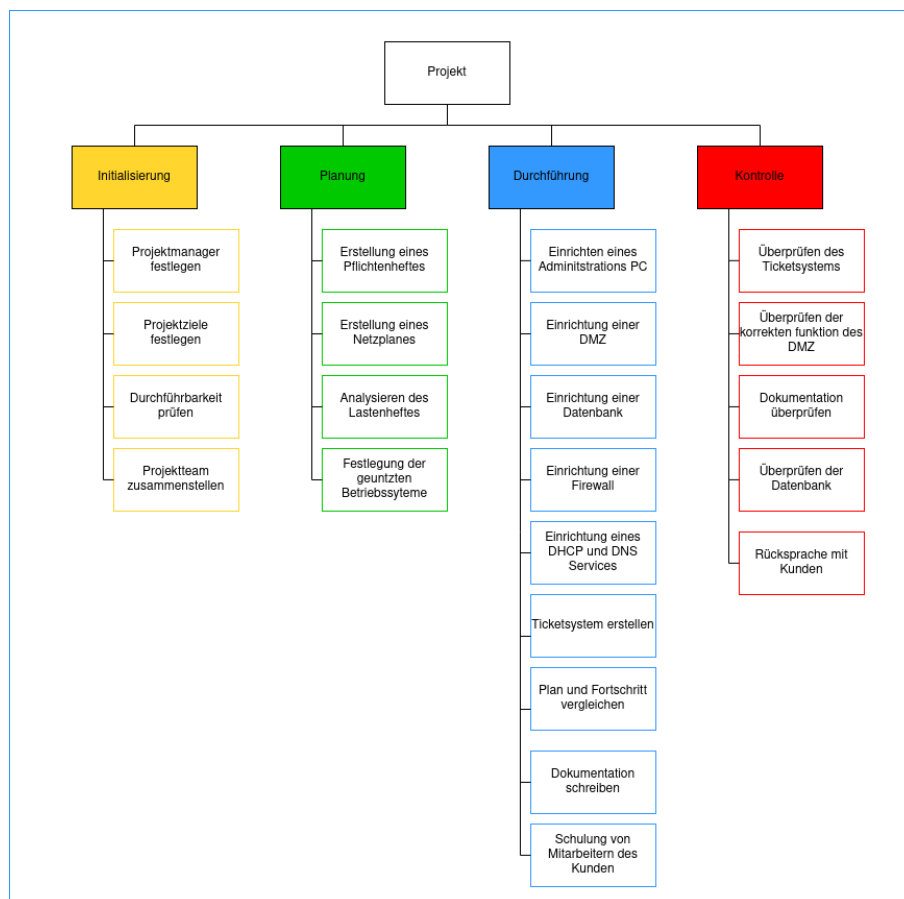


Abbildung 1.2: Projektstrukturplan

GANTT Diagramm Gruppe 7

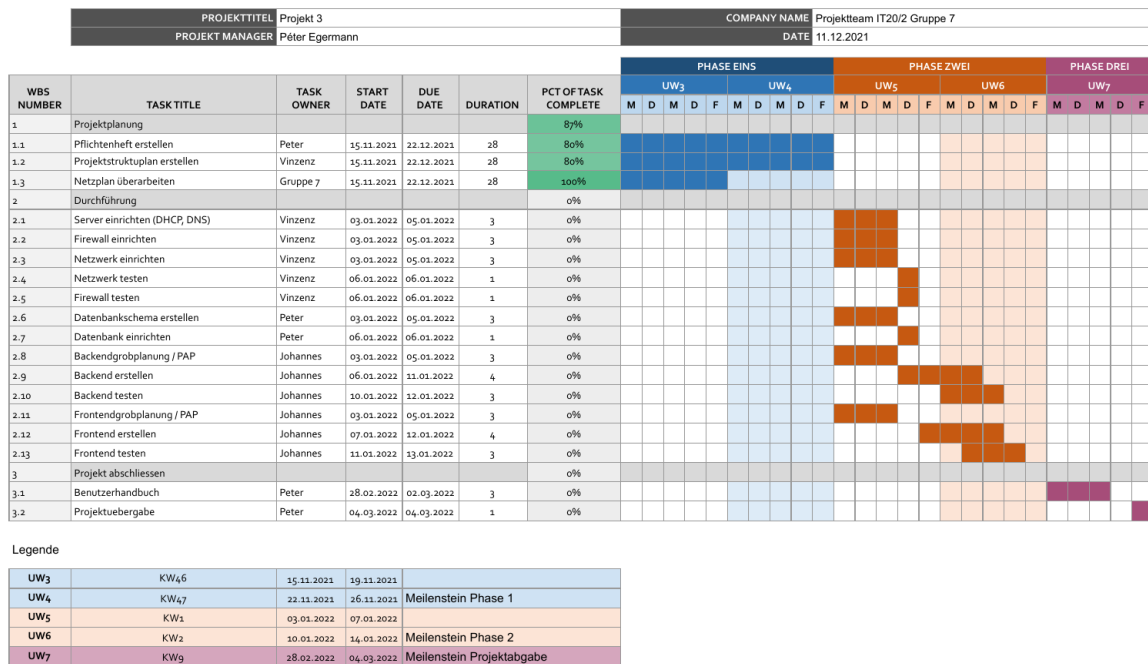


Abbildung 1.3: Gantt-Diagramm

1.10 Kosten-Nutzen-Analyse

Eine Kosten-Nutzen-Analyse ist zum jetzigen Zeitpunkt nicht notwendig, da der Support erst mal entlastet werden muss. Dies ist durch das neue System auf jeden Fall der Fall, da quasi der Kunde das Ticket erstellt und nicht der Support-Mitarbeiter. Somit kann sich voll auf das Beheben des Problems konzentriert werden.

2 Auswertung und Reflexion

2.1 Ablaufdokumentation

In diesem Projekt werden drei virtuelle Netze eingerichtet, um den Aufbau eines echten Netzwerks zu simulieren. Dabei handelt es sich um das rote Netz, das das Internet wiedergeben soll, das orange Netz, was eine sogenannte DMZ darstellt und das grüne Netz, welches das interne Netz darstellt.

Um die Kommunikation und den Zugriff zwischen den Netzen zu regeln wird in diesem Projekt die Firewall verwendet. Somit kann aus dem roten Netz nur mit dem orangen Netz kommuniziert werden, aus dem grünen Netz ist kein Zugriff auf das Internet möglich und zwischen dem orangen und grünen Netz sind nur bestimmte Ports zur Kommunikation und Datenübertragung zugelassen.

Um Maschinen in den verschiedenen Netzen darzustellen wurden vier verschiedene virtuelle Maschinen (VMs) aufgesetzt, die bis auf der IPFire auf CentOS 8 Stream basieren:

- IPFire (als Knotenpunkt für alle drei Netze)
- Admin-PC (grünes Netz)
- DHCP-DNS-DB-Server (grünes Netz)
- Webserver (DMZ, oranges Netz)

2.2 Einrichtung IPFire

Vor dem Einrichten der IPFire müssen noch zwei Netzwerke zu dem durch VMWare standardmäßig schon bestehenden Netzwerk hinzugefügt werden, da die Firewall mit drei verschiedenen Netzen interagieren soll. Dazu werden den einzelnen Netzen verschiedene MAC-Adressen zugeteilt, damit sie in der späteren Nutzung zuordenbar sind.

Die Netzwerke werden wie in Tabelle 2.1 zu sehen verteilt.

Tabelle 2.1: Netzwerkauslegung

Netzwerk-Farbe	MAC-Adresse	Netzwerk
Rot	00:50:56:32:BA:0F	NAT
Grün	00:50:56:3D:EC:D6	VMnet1
Orange	00:50:56:3E:56:B7	VMnet2

Als Hostname der Firewall wird „ipfire“ und als Domaine „doubtful-joy07.com“ festgelegt. Nach dem Auswählen der Sprache wird aufgrund der Kundenspezifikation das Filesystem „ext4 Filesystem“ ausgewählt. Nach dem Zuweisen der einzelnen Netze mit den IP- und MAC-Adressen

wird der DHCP deaktiviert, damit es später mit dem DHCP-Server im grünen Netz nicht zu Komplikationen führt.

Nach dem Einrichten der Firewall wird aus dem grünen Netz mittels des IPFire-Webinterfaces verschiedene Einstellungen der Firewall bearbeitet. Für die Verbindungen zwischen Admin-PC, DHCP-DNS-Datenbank (DB)-Server und Webserver sowie für die Erreichbarkeit des Webservers aus dem roten Netz werden vier verschiedene Regeln erstellt, die in Abb. 2.1 zu sehen sind.

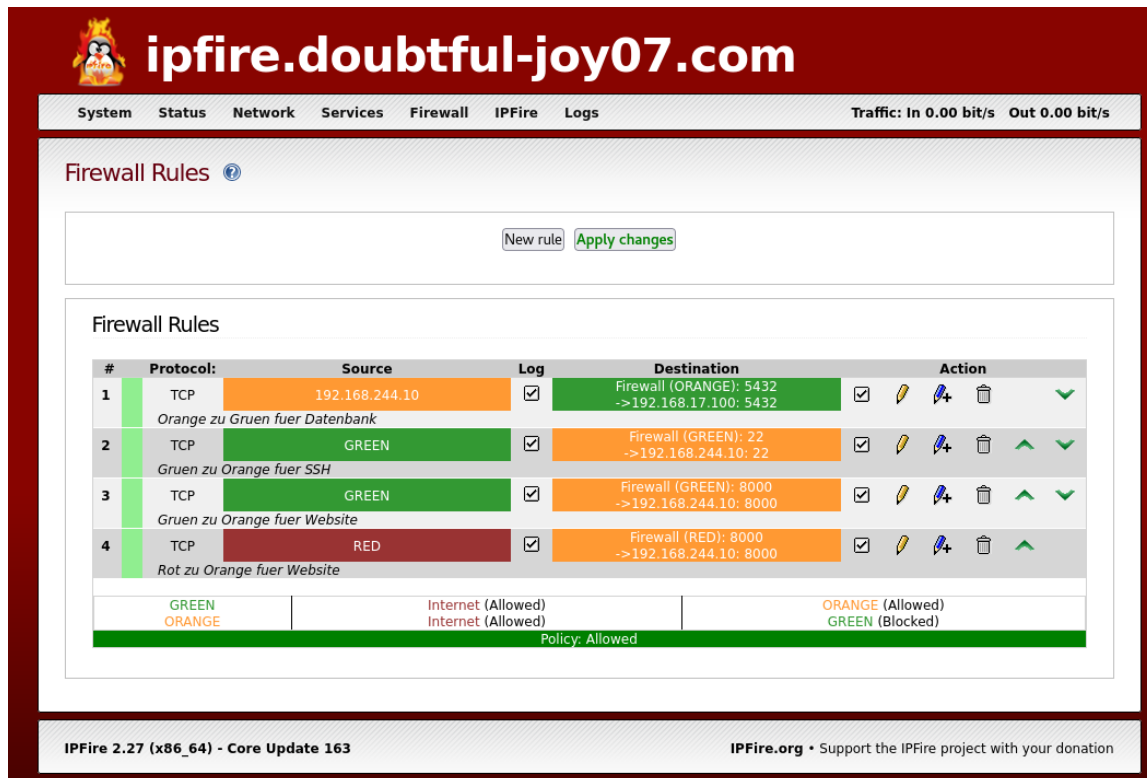


Abbildung 2.1: Firewall-Regeln

Die erste Regel dient der Kommunikation zwischen Webserver (DMZ) und Datenbank (grünes Netz), sodass Tickets abgerufen und gespeichert werden können. Damit aus dem grünen Netz der Webserver gestartet und gestoppt werden kann, wird Regel 2 implementiert, welche einen Secure Shell (SSH)-Zugang aus dem grünen Netz erlaubt. Die Erreichbarkeit des Webinterface der Ticket-Seite durch die Mitarbeiter aus dem grünen Netz wird mit Regel 3 erreicht. Die letzte Regel erlaubt den Zugriff aus dem Internet auf die Ticket-Website.

2.3 Einrichtung Admin-PC

Nach der Standard-Installation von CentOS 8 Stream wird das Netzwerk der VM angepasst. Hier wird der DNS auf die IP-Adresse des DHCP-DNS-DB-Servers gesetzt.

Da in CentOS 8 Stream SSH-Client und -Server bereits installiert und aktiviert sind, kann der Webserver direkt angesprochen werden. Dies erfolgt über das Gateway des grünen Netzes, wie in Abb. 2.2 auf der nächsten Seite zu sehen. Um zu sehen, ob der Webserver aktiv ist, können mittels `ps -ef | grep python` alle laufenden Python-Anwendungen aufgelistet werden, was

ebenfalls in Abb. 2.2 zu sehen ist.

```
[admin@localhost ~]$ ssh admin@192.168.17.3
admin@192.168.17.3's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sun Mar 13 10:01:19 2022 from 192.168.17.58
[admin@localhost ~]$ ps -ef | grep python
root      1076      1   0 09:22 ?        00:00:00 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid
root      1144      1   0 09:22 ?        00:00:04 /usr/libexec/platform-python -Es /usr/sbin/tuned -l -P
admin     5027      1   0 10:02 ?        00:00:00 python3.10 main.py
admin     5139    5096   0 10:03 pts/1    00:00:00 grep --color=auto python
[admin@localhost ~]$
```

Abbildung 2.2: SSH-Login sowie Auflisten aller laufenden Python-Anwendungen

Ist der Webserver aktiv und soll gestoppt werden, kann mittels `ps -ef | grep python` die ID des Scripts ermittelt und mittels `kill -9 ID` gestoppt werden. Dies ist in Abb. 2.3 zu sehen.

```
[admin@localhost ~]$ kill -9 5027
[admin@localhost ~]$ ps -ef | grep python
root      1076      1   0 09:22 ?        00:00:00 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid
root      1144      1   0 09:22 ?        00:00:04 /usr/libexec/platform-python -Es /usr/sbin/tuned -l -P
admin     5172    5096   0 10:04 pts/1    00:00:00 grep --color=auto python
[admin@localhost ~]$
```

Abbildung 2.3: Stoppen einer bestimmten Python-Anwendungen

Soll der Webserver gestartet werden, kann dies mittels Navigation in den Ordner, in dem die auszuführende Datei liegt und `python3.10 name-der-datei &` gestartet werden, zu sehen in der Abb. 2.4. Das `&` erlaubt das Laufen der Anwendung im Hintergrund und wird so nicht gestoppt, wenn die SSH-Verbindung geschlossen wird.

```
[admin@localhost ~]$ cd /home/admin/Documents/LF9_Project3-main/backend/
[admin@localhost backend]$ python3.10 main.py &
[1] 5027
[admin@localhost backend]$ INFO: Started server process [5027]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
exit
logout
Connection to 192.168.17.3 closed.
[admin@localhost ~]$
```

Abbildung 2.4: Starten einer bestimmten Python-Anwendungen

2.4 Einrichtung DHCP-DNS-DB-Server

2.4.1 Einrichtung der CentOS-Installation

Wie auch der Admin-PC wird der DHCP-DNS-DB-Server mit CentOS 8 Stream eingerichtet. In den Netzwerkeinstellungen wird die IP-Adresse fest auf 192.168.17.100, die Netzwerkmaske auf 255.255.255.0 und das Gateway auf 192.169.17.3 gesetzt. Außerdem werden die Dienste DNS und DHCP in der Firewall freigegeben, zu sehen in Listing 2.1.

Listing 2.1: Dienste-Freigabe einer CentOS-Firewall

```
# firewall-cmd --add-service=dns --permanent
```

```
# firewall-cmd --add-service=dhcp --permanent
# firewall-cmd --reload
```

2.4.2 Einrichtung des DHCP und DNS

Die Installation und Einrichtung des DHCPs und DNS erfolgt mittels der Anleitung für dnsmasq von Michelle Ferron. [2]

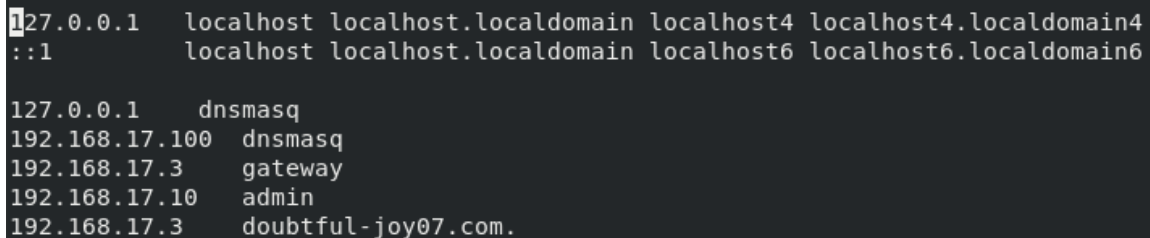
Die Einstellungen des DHCP wird mittels der in Listing 2.2 zu sehenden Befehle und Einstellungen angepasst.

Listing 2.2: DHCP-Einstellungen

```
# nano /etc/dnsmasq.conf
listen-address=::1,127.0.0.1,192.168.17.100
interface=ens160
domain=doubtful-joy07.com
```

Nach der Einrichtung wird das Netzwerk des Admin-PCs ausgeschaltet, die DHCP-Einstellungsoption ausgewählt und das Netzwerk wieder angeschaltet. Der Admin-PC bekommt durch den DHCP automatisch eine neue IP-Adresse.

Die Einstellungen des DNS wird mittels der `/etc/hosts`-Datei angepasst. Die hier einzustellenden Werte sind in Abb. 2.5 zu sehen.

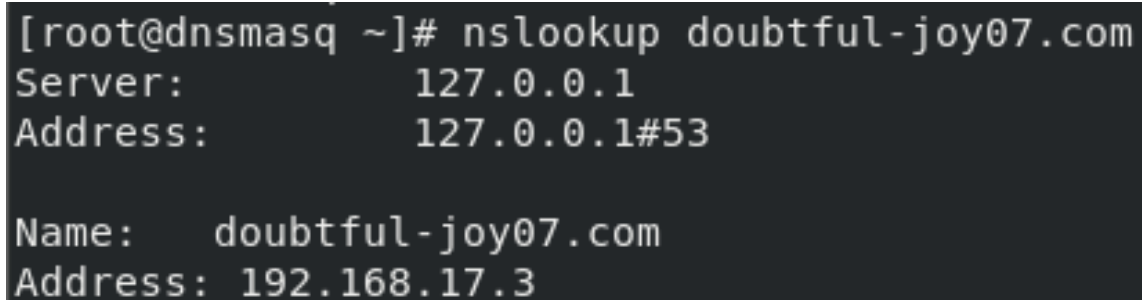


```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6

127.0.0.1    dnsmasq
192.168.17.100 dnsmasq
192.168.17.3  gateway
192.168.17.10 admin
192.168.17.3  doubtful-joy07.com.
```

Abbildung 2.5: DNS-Einstellungen

Die Funktion des DNS kann mittels `nslookup`, zu sehen in Abb. 2.6, oder über das Aufrufen des Frontends des Webservers, zu sehen in Abb. 2.7 auf der nächsten Seite, überprüft werden.



```
[root@dnsmasq ~]# nslookup doubtful-joy07.com
Server:          127.0.0.1
Address:         127.0.0.1#53

Name:   doubtful-joy07.com
Address: 192.168.17.3
```

Abbildung 2.6: Überprüfen des DNS mittels Namens

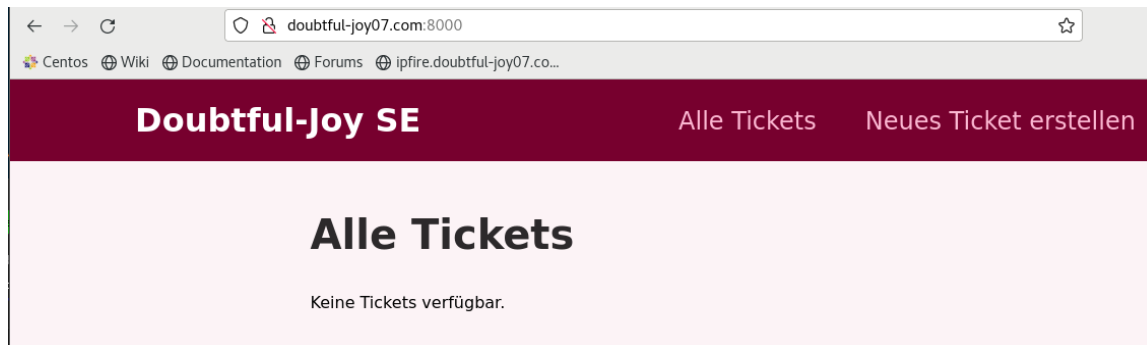


Abbildung 2.7: Aufrufen des Frontends mittels Name

2.4.3 Einrichtung des Datenbank-Servers

Die Einrichtung eines Datenbank-Servers erfolgt mittels der Anleitung von Aaron Kili. [3] Mit dieser wird PostgreSQL, ein „leistungsstarkes, weit verbreitetes, quelloffenes, plattformübergreifendes und fortschrittliches objektrelationales Datenbanksystem“ [3] und pgAdmin, ein „fortschrittliches, quelloffenes, voll funktionsfähiges und webbasiertes Verwaltungs- und Managementwerkzeug“ [3], installiert.

Mittels des Webinterfaces, das pgAdmin bereitstellt, lässt sich eine Datenbank erstellen, zu sehen in Abb. 2.8 auf der nächsten Seite. Der Name der Datenbank wird auf „tickets“, der Benutzername auf „ticketadmin“ und das Passwort auf „adminadmin“ festgelegt.

Um die Kommunikation zwischen der Datenbank und dem Backend des Webservers zu erlauben, muss der Port 5432 freigegeben werden, zu sehen in Listing 2.3.

Listing 2.3: Port-Freigabe einer CentOS-Firewall

```
# firewall-cmd --zone=public --add-port=5432/tcp --permanent
# firewall-cmd --reload
```

Ebenso muss die `postgresql.conf` angepasst werden, so dass ein Zugriff von außerhalb überhaupt möglich ist. [1] Dies ist in folgendem Listing 2.4 zu sehen.

Listing 2.4: Einrichtung Zugriff PostgreSQL

```
# nano /var/lib/pgsql/data/postgresql.conf
listen_addresses = '*'
```

Da im Front- und Backend Timestamps als Zahlen verwendet werden, muss die Tabelle noch geändert werden, damit kein `integer-out-of-range`-Fehler geworfen wird. Dies wird mittels der Befehls `ALTER TABLE` wie in Listing 2.5 umgesetzt.

Listing 2.5: Ändern der tickets-Tabelle

```
ALTER TABLE order_detail ALTER COLUMN amount TYPE BIGINT;
```

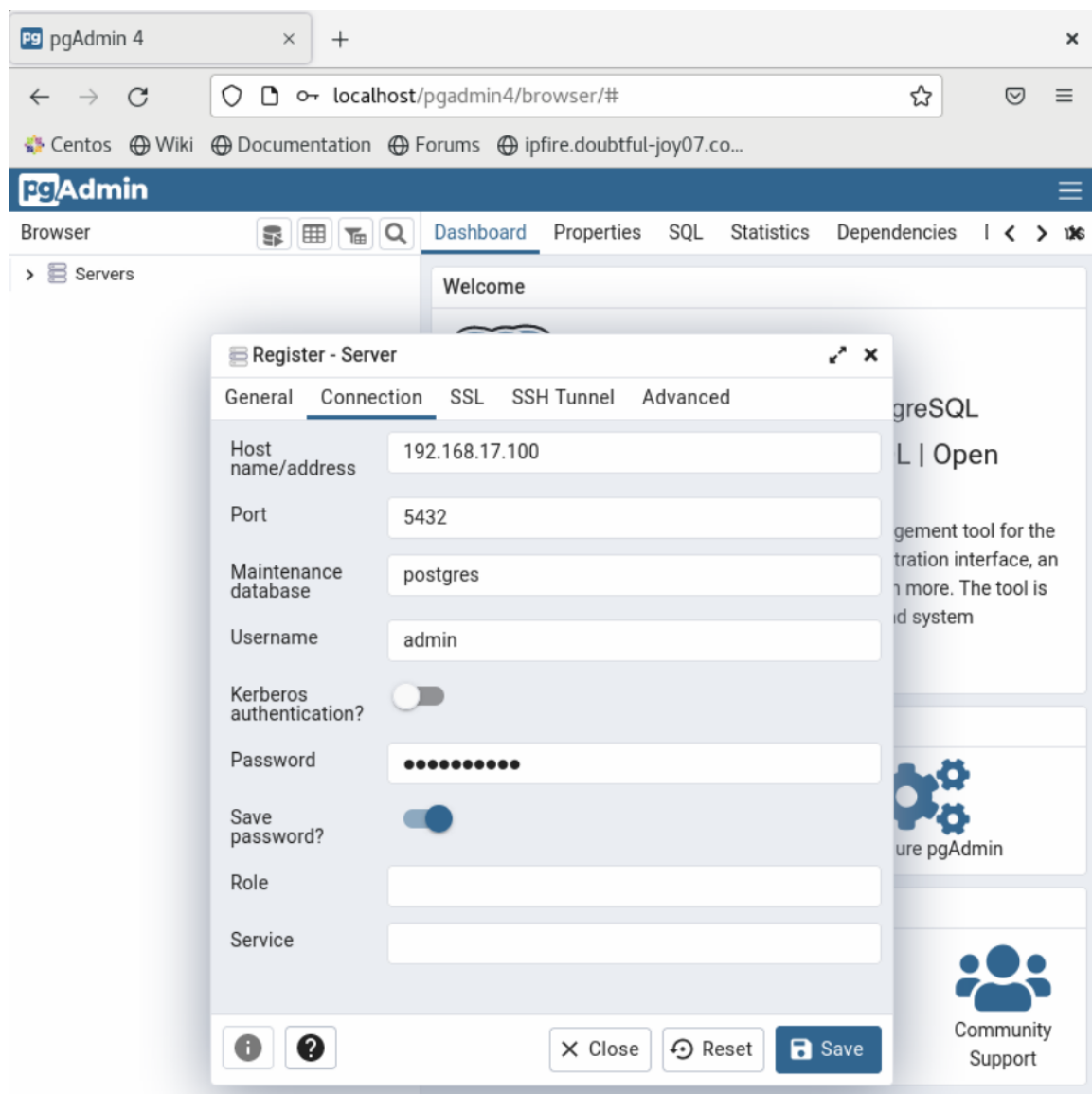


Abbildung 2.8: Einrichtung der Tickets-Datenbank

2.5 Einrichtung Webserver

2.5.1 Einrichtung der CentOS-Installation

Nach der Standard-Installation von CentOS 8 Stream wird das Netzwerk der VM angepasst. Hier wird die IP-Adresse fest auf 192.168.244.10, die Netzwerkmaske auf 255.255.255.0 und das Gateway auf 192.169.244.3 gesetzt. Außerdem wird der Port 8000 und Port 22 freigeben. Port 8000 dient dem Zugriff auf die Website von den beiden anderen Netzen aus und Port 22 erlaubt den SSH-Zugriff des Admin-Pcs. Als Beispiel ist die Port-Freigabe für Port 8000 im Listing 2.6 zu sehen.

Listing 2.6: Port-Freigabe einer CentOS-Firewall

```
# firewall-cmd --zone=public --add-port=8000/tcp --permanent
# firewall-cmd --reload
```

Abschließend wird Python3.10 installiert, um das Backend betreiben zu können.

2.5.2 Einrichtung des Backends

Das Backend wird mit Python3.10 umgesetzt. Hierfür wird mit Hilfe von FastAPI eine Application Programming Interface (API) geschrieben, die verschiedene Routen bereitstellt, mit denen create, read, update und delete (CRUD)-Anweisungen ausgeführt werden können. Außerdem dient das Backend auch gleich als Server für das Frontend, da es eben dieses bereitstellt.

Die benötigten Pakete können mittels `pip3.10 requirements.txt` installiert werden.

2.5.3 Einrichtung des Frontends

Für das Frontend wird die JavaScript-Softwarebibliothek React verwendet. Hier werden alle vom Kunden geforderten Anzeige und Bedienelemente implementiert. Da das Frontend bereits nach der Umsetzung gebaut und durch das Backend bereitgestellt wird, müssen keine Pakete installiert werden.

2.6 Soll-Ist-Vergleich

Der Zustand der abgelieferten Arbeit entspricht dem Soll-Zustand und somit den Kundenwünschen. Es werden alle Kriterien umgesetzt. Das Firewall-System, DHCP und DNS, Webserver und Datenbanksystem funktionieren einwandfrei.

2.7 Abweichung zum Zeitplanung

Der ursprüngliche Zeitplan, zu sehen in Abb. 1.3 auf Seite 6, bzw. Abb. A.1 auf Seite 20, konnten nicht eingehalten werden. Es gab zwei Probleme:

- Die Linux-Clients im IPFire Netz konnten keine Software installieren.
Hier wurde vermutet, dass es an der Einrichtung der Firewall-Regeln lag, weswegen eine komplette Neuinstallation vorgenommen wurde. Dies hat unnötig Zeit gefressen. Die Lösung war dann, die System zu Hause einzurichten.

- Durch den Zeitverzug, den das erste Problem mit sich trug, konnte die Dokumentation nicht rechtzeitig fertiggestellt werden.

Das aktualisierte Gantt-Diagramm ist in klein in Abb. 2.9 und groß im Anhang in Abb. A.2 auf Seite 21 zu sehen.

GANTT Diagramm Gruppe 7

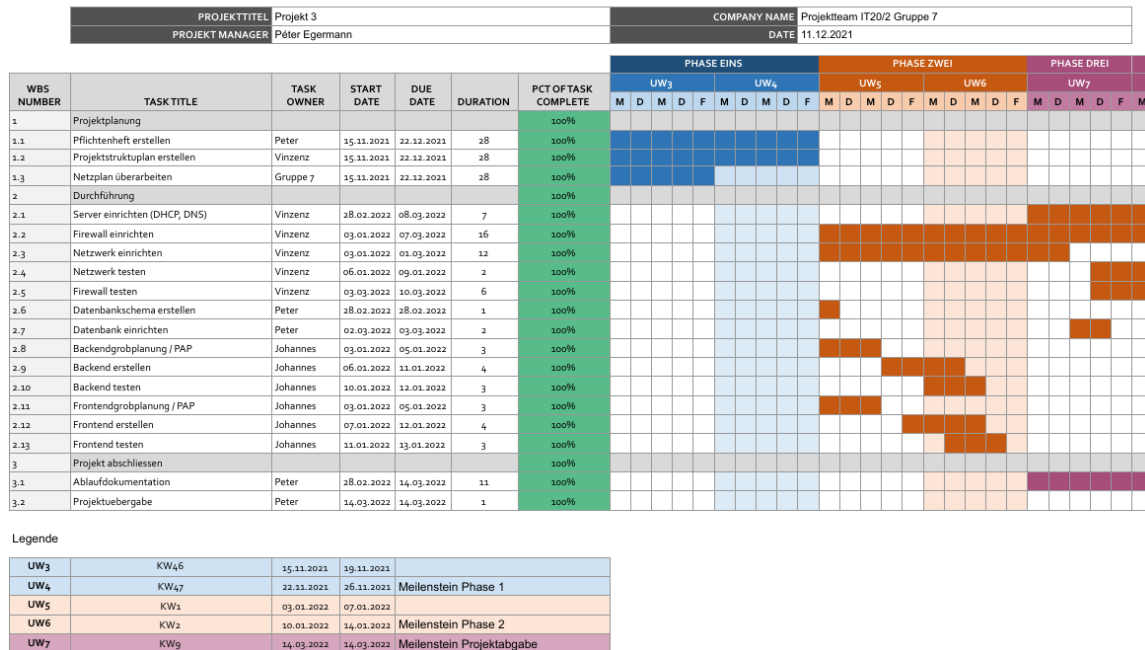


Abbildung 2.9: Gantt-Diagramm

2.8 Optimierungsvorschläge zur Projektrealisierung

Das Projektteam IT20/2 Gruppe 7 besteht aus drei Anwendungsentwicklern, was die Sache deutlich erschwert hat. Hier wäre eine Überarbeitung der Gruppeneinteilung von Vorteil gewesen, so dass ein Team durch einen Systemintegrator und einen Anwendungsentwickler gebildet wird, wodurch sich gewisse Synergieeffekte ergeben könnten.

Abbildungsverzeichnis

1.1	Schnittstellenspezifikation	4
1.2	Projektstrukturplan	5
1.3	Gantt-Diagramm	6
2.1	Firewall-Regeln	8
2.2	SSH-Login sowie Auflisten aller laufenden Python-Anwendungen	9
2.3	Stoppen einer bestimmten Python-Anwendungen	9
2.4	Starten einer bestimmten Python-Anwendungen	9
2.5	DNS-Einstellungen	10
2.6	Überprüfen des DNS mittels Namens	10
2.7	Aufrufen des Frontends mittels Name	11
2.8	Einrichtung der Tickets-Datenbank	12
2.9	Gantt-Diagramm	14

Tabellenverzeichnis

1.1	Ansprechpartner Auftraggeber	1
1.2	Ansprechpartner Auftragnehmer	1
2.1	Netzwerkauslegung	7

Listings

2.1	Dienste-Freigabe einer CentOS-Firewall	9
2.2	DHCP-Einstellungen	10
2.3	Port-Freigabe einer CentOS-Firewall	11
2.4	Einrichtung Zugriff PostgreSQL	11
2.5	Ändern der tickets-Tabelle	11
2.6	Port-Freigabe einer CentOS-Firewall	13

Literaturverzeichnis

- [1] Karim Buzdar. *How to install PostgreSQL Database Server CentOS 8*. Vitux. Verfügbar unter: <https://vitux.com/postgresql-centos/>. abgerufen am 14.03.2022.
- [2] Michele Ferron. *Installing DNS Server on CentOS/RHEL using dnsmasq*. Zextras Suite & Zimbra OSE. 2021. Verfügbar unter: <https://community.zextras.com/dns-server-installation-guide-on-centos-7-rhel-7-and-centos-8-rhel-8-using-dnsmasq/>. abgerufen am 14.03.2022.
- [3] Aaron Kili. *How to Install PostgreSQL and pgAdmin in CentOS 8*. TecMint. 2021. Verfügbar unter: <https://www.tecmint.com/install-postgressql-and-pgadmin-in-centos-8/>. abgerufen am 14.03.2022.

Anhang

A Gantt-Diagramme

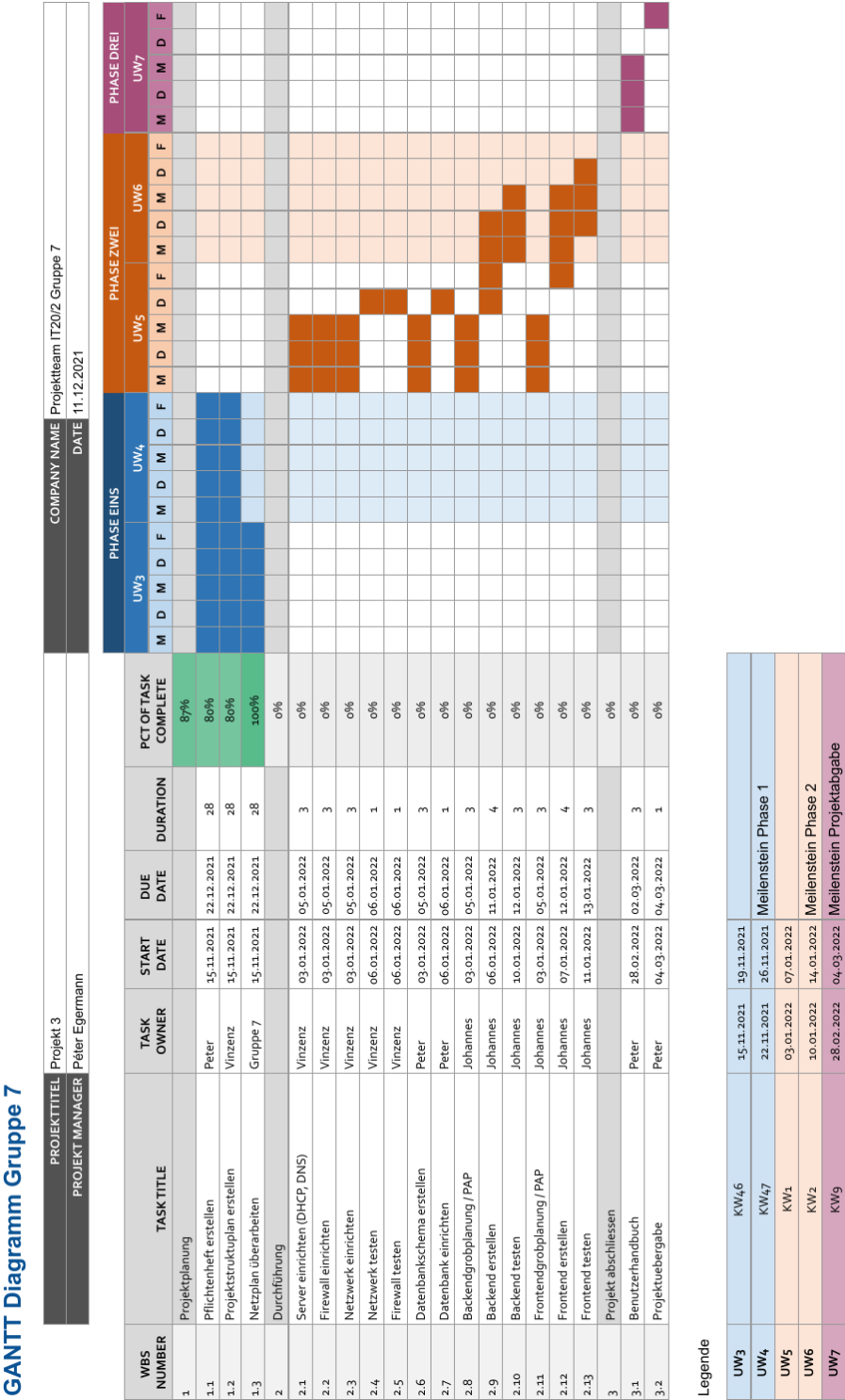


Abbildung A.1: Gantt-Diagramm

GANTT Diagramm Gruppe 7

PROJEKTITITEL			PROJEKT MANAGER		Projekt 3		COMPANY NAME		Projektteam IT20/2 Gruppe 7																					
					Péter Egernmann				DATE																					
WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	PHASE INS						PHASE ZWEI						PHASE DREI											
							UW3			UW4			UW5			UW6			UW7											
							M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M	D	M	D	F	M	D	F	M
1	Projektplanung					100%																								
1.1	Pflichtenheft erstellen	Peter	15.11.2021	22.12.2021	28	100%																								
1.2	Projektstrukturplan erstellen	Vinzenz	15.11.2021	22.12.2021	28	100%																								
1.3	Netzplan überarbeiten	Gruppe 7	15.11.2021	22.12.2021	28	100%																								
2	Durchführung					100%																								
2.1	Server einrichten (DHCP, DNS)	Vinzenz	28.03.2022	08.03.2022	7	100%																								
2.2	Firewall einrichten	Vinzenz	03.01.2022	07.03.2022	16	100%																								
2.3	Netzwerk einrichten	Vinzenz	03.01.2022	01.03.2022	12	100%																								
2.4	Netzwerk testen	Vinzenz	06.01.2022	09.01.2022	2	100%																								
2.5	Firewall testen	Vinzenz	03.03.2022	10.03.2022	6	100%																								
2.6	Datenbankschema erstellen	Peter	28.03.2022	28.03.2022	1	100%																								
2.7	Datenbank einrichten	Peter	02.03.2022	03.03.2022	2	100%																								
2.8	Backendgroßplanung / PAP	Johannes	03.01.2022	05.01.2022	3	100%																								
2.9	Backend erstellen	Johannes	06.01.2022	11.01.2022	4	100%																								
2.10	Backend testen	Johannes	10.01.2022	12.01.2022	3	100%																								
2.11	Frontendgroßplanung / PAP	Johannes	03.01.2022	05.01.2022	3	100%																								
2.12	Frontend erstellen	Johannes	07.01.2022	12.01.2022	4	100%																								
2.13	Frontend testen	Johannes	11.01.2022	13.01.2022	3	100%																								
3	Projekt abschliessen					100%																								
3.1	Ablaufdokumentation	Peter	28.02.2022	14.03.2022	11	100%																								
3.2	Projektuebergabe	Peter	14.03.2022	14.03.2022	1	100%																								
Legende																														
UW3	KW46		15.11.2021	19.11.2021																										
UW4	KW47		22.11.2021	26.11.2021																										
UW5	KW1		03.01.2022	07.01.2022																										
UW6	KW2		10.01.2022	14.01.2022																										
UW7	KW9		14.03.2022	14.03.2022																										
Meilenstein Projektlaababe																														

Abbildung A.2: Update Gantt-Diagramm

B Netzwerkplan

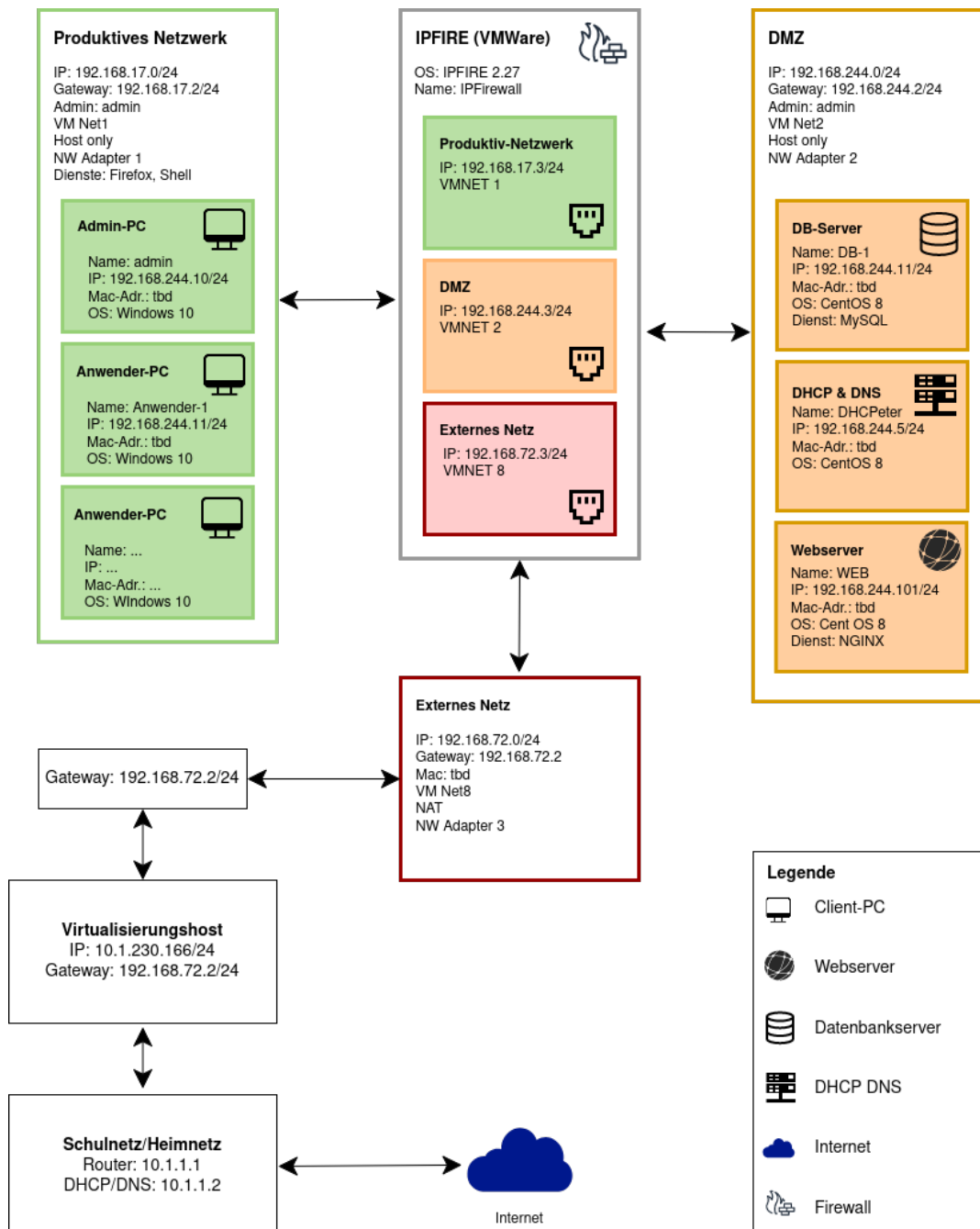


Abbildung B.1: Netzwerkplan