|  |  |
| --- | --- |
|  | |
| Gruppenmitglieder: | Marvin Madeira  Felix Rünzler |

|  |  |  |  |
| --- | --- | --- | --- |
| Versionsnummer | Datum | Änderungen | Bearbeiter |
| 1 | 31.01.2024 | Planung Netzwerk, Netzplan | Marvin, Felix |
| 2 | 07.02.2024 | Kostenaufstellung, Geräte ausgesucht | Marvin, Felix |
| 3 | 21.02.2024 | Packet Tracer ausarbeitung, Mitarbeiteranlegen | Felix,  Marvin |
| 4 | 28.02.2024 | Packet Tracer Ausarbeitung, Datenbanken | Marvin, Felix |
| 5 | 06.03.2024 | Packet Tracer fertig, C# und Datenbanken | Marvin, Felix |
| 6 | 13.03.2024 | C# Programm mit UI, Database Connector und Datenbank fertig | Marvin, Felix |
| 7 | 20.03.2024 |  | Marvin, Felix |
|  |  |  |  |

# Projektmanagement

# Teil 1: Planung Netzwerk

### Anforderungsanalyse



<Beschreibung der wesentlichen Anforderungen an das Netzwerk, aus dem dann die Umsetzung abgeleitet werden kann. Zum Beispiel: Büro soll X Rechner haben, daraus ergibt sich ein Subnet Y>

Räumlichkeit: 200qm pro Stockwerk (direkt übereinander)

Mitarbeiter (mindestens 30) per LAN -> 64er Netz DHCP für jede Etage

IP-Telefonie für alle -> 64er Netz pro Stockwerk statische Adressen

2 Konferenzräume (1 pro Stockwerk)

IP-Telefonie für alle

2 Drucker (1 pro Stockwerk)

Pro User 1 TB Speicherplatz (active directory oder so)

Serverraum: Switch, Router, Server, Speicherkapazität

1 öffentliche IP

Budget 75.000€

Gastzugang per WLAN (evtl Repeater oder Access Points) -> 64er Netz pro Stockwerk dyn.

Aktuell NAT



Teams und Telefonie getrennt

Arbeitsplatz: 1 Bildschirm, Docking Station, Tastatur, LAN-Anschluss, Maus,

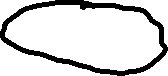
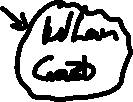
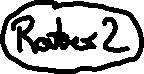
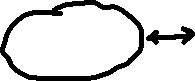
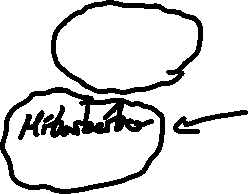
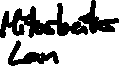
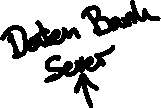
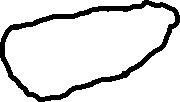
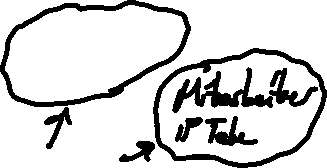
### Abgrenzungskriterien

*<Funktionalität die das System nicht besitzt>*



Externe Websites

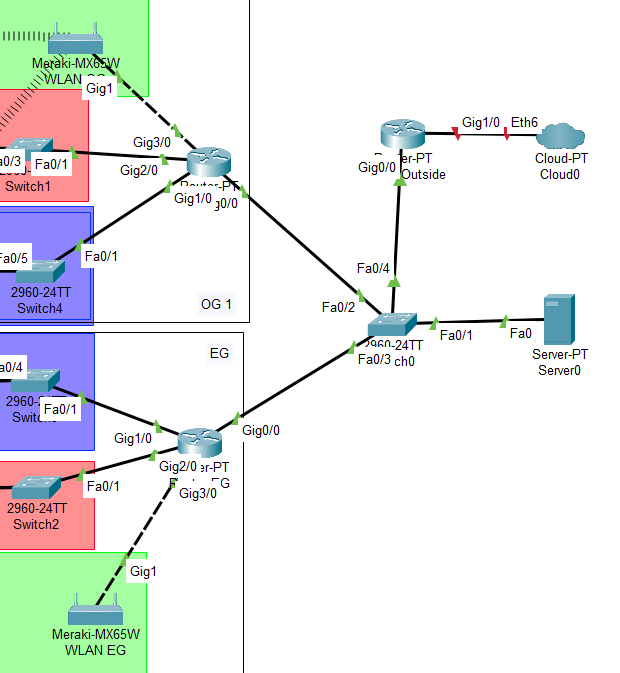
Kabelkanäle



### Netzplan

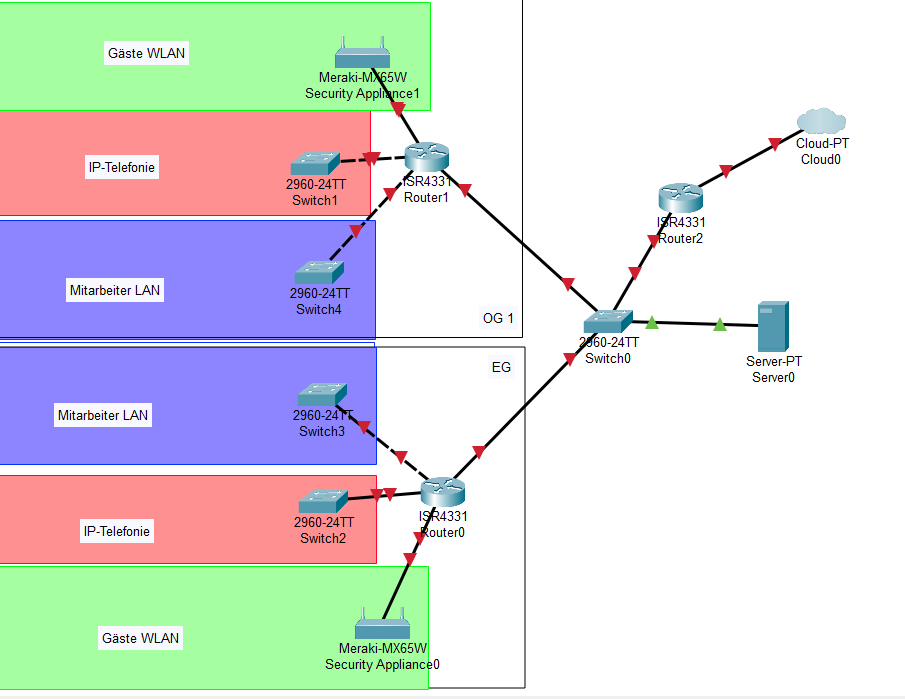
*<Tabellarische Auflistung der geplanten Netze als Vorlage für technische Umsetzung, d.h. Netze mit Subnetmaske, Gerätebezeichnungen für Router und Switche, Tabelle mit Verbindungen zwischen Backbone Geräten, also zum Beispiel switch A – Port 4 auf Router B – Interface FA/0>*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Netzwerk | Netz-ID | | Subnetz-maske | Anzahl IPs | | Nutzbarer Adressbereich | | Gateway |
| Mitarbeiter LAN oben | 192.168.0.0 | | 26 | 62 | | 192.168.0.1 - 192.168.0.62 | | 192.168.0.1 |
| Mitarbeiter LAN unten | 192.168.0.64 | | 26 | 62 | | 192.168.0.65 - 192.168.0.126 | | 192.168.0.65 |
| Gast WLAN oben | 192.168.0.128 | | 26 | 62 | | 192.168.0.129 - 192.168.0.190 | | 192.168.0.129 |
| Gast WLAN unten | 192.168.0.192 | | 26 | 62 | | 192.168.0.193 - 192.168.0.254 | | 192.168.0.193 |
| IP-Telefone oben | 192.168.1.0 | | 26 | 62 | | 192.168.1.1 - 192.168.1.62 | | 192.168.1.1 |
| IP-Telefone unten | 192.168.1.64 | | 26 | 62 | | 192.168.1.65 - 192.168.1.126 | | 192.168.1.65 |
| Intern zw. Routern und Server | 192.168.1.128 | | 29 | 6 | | 192.168.1.129 - 192.168.1.134 | | 192.168.1.129 |
| Intern Router <-> Internet | 192.168.1.192 | | 30 | 2 | | 192.168.1.193 - 192.168.1.194 | | 192.168.1.193 |
| Gerät | Interface | IP-Adresse | | | Netzwerk | | Verbundenes Interface | | |
| Router Outside | Gig 0/0 | 192.168.1.129 | | | Intern zw. Routern und Server | | Switch Intern Fa 0/4 | | |
|  | Gig 1/0 |  | | |  | |  | | |
| Router OG | Gig 0/0 | 192.168.1.130 | | | Intern zw. Routern und Server | | Switch Intern Fa 0/2 | | |
|  | Gig 1/0 | 192.168.0.1 | | | Mitarbeiter LAN oben | | Switch Mitarbeiter LAN OG Fa 0/1 | | |
|  | Gig 2/0 | 192.168.1.1 | | | IP-Telefone oben | | Switch IP Telefonie OG Fa 0/1 | | |
|  | Gig 3/0 | 192.168.0.129 | | | Gast WLAN oben | | Meraki WLAN Gig1 | | |
| Router EG | Gig 0/0 | 192.168.1.131 | | | Intern zw. Routern und Server | | Switch Intern Fa 0/3 | | |
|  | Gig 1/0 | 192.168.0.65 | | | Mitarbeiter LAN unten | | Switch Mitarbeiter LAN EG Fa 0/1 | | |
|  | Gig 2/0 | 192.168.1.65 | | | IP-Telefone unten | | Switch IP Telefonie EG Fa 0/1 | | |
|  | Gig 3/0 | 192.168.0.193 | | | Gast WLAN unten | | Meraki WLAN Gig1 | | |
| Server | Fa 0/0 | 192.168.1.132 | | | Intern zw. Routern und Server | | Switch Intern Fa 0/1 | | |
| Drucker OG | 0 | 192.168.0.2 | | | Mitarbeiter LAN oben | | Switch Mitarbeiter LAN OG Fa 0/2 | | |
| Drucker EG | 0 | 192.168.0.66 | | | Mitarbeiter LAN unten | | Switch Mitarbeiter LAN EG Fa 0/2 | | |



### Demonstrationsprototyp Packet Tracer

<Aufbau eines exemplarischen, funktionsfähigen Netzwerks>

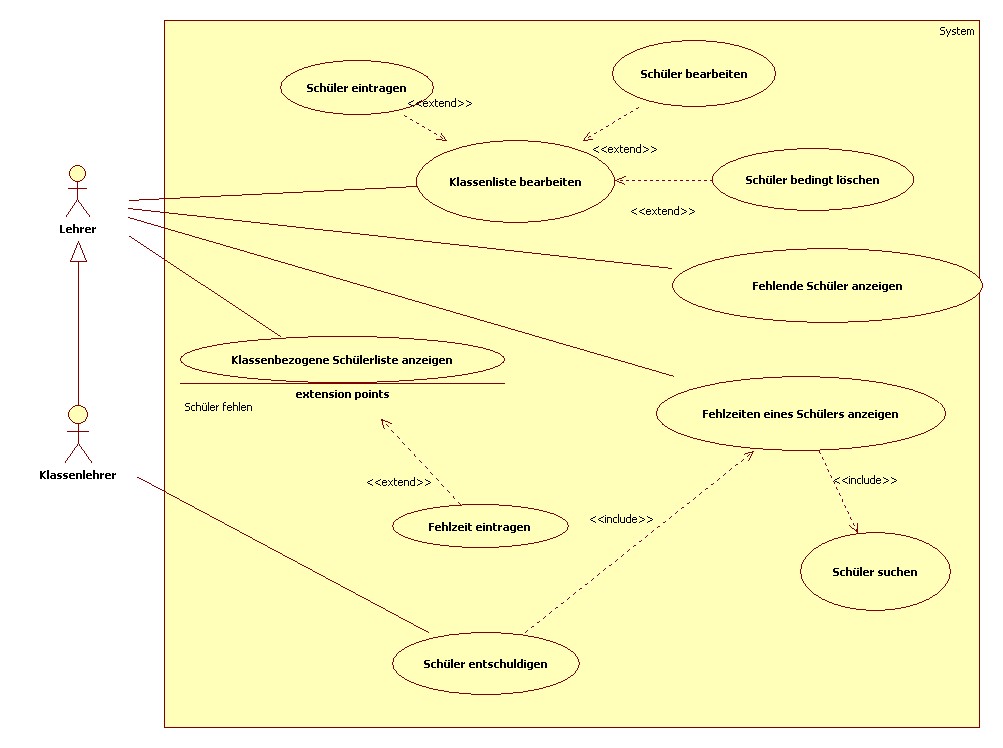


# Teil 2: Datenbank mit Datenexport

### 1. Anforderungsanalyse

### Use Case Analyse

<*Grafische Darstellung der Anforderungen in Form von Use Cases>*



### Use Case Beschreibung

*<Detaillierte Beschreibung der Use Cases nach einer vorgegebenen Struktur>*

|  |  |
| --- | --- |
|  | |
| Primärer Aktor: |  |
| Vorbedingung: |  |
| Wichtigstes Erfolgsszenario: |  |
| Wichtige Varianten: | 2.a Fehlzeit befindet sich nicht im System obwohl Entschuldigung vorliegt |
| Auswirkungen: | Fehlzeit als ‚Entschuldigt’ markiert |
| Anmerkungen: | Klassenlehrer kann sowohl Anfang als auch Ende der Fehlzeit bearbeiten |
| Offene Fragen: | Darf vorhandene Fehlzeit gelöscht werden? |

|  |  |
| --- | --- |
| USE CASE: Klassenliste bearbeiten | |
| Primärer Aktor | Lehrer |
| Vorbedingung | Klassenliste nicht aktuell |
| Wichtigstes Erfolgsszenario: | Lehrer wählt Klasse aus |
| Wichtige Varianten: | 2.a keine Schüler vorhanden |
| Auswirkungen: | Schülerliste wird aktualisiert |
| Anmerkungen: | Jeder Lehrer kann Klassenliste jederzeit bearbeiten |
| Offene Fragen: | - |

### Abgrenzungskriterien

*<Funktionalität die das System nicht besitzt>*

# Systemarchitektur

*<Beschreibung der verwendeten Infrastrukturkomponenten (Datenbanken, Server, Technologien) und grafische Darstellung der Gesamtarchitektur, z.B* C#, MySQL Datenbank;

*>*

# Implementierung

### Statische Perspektive (Klassendiagramme)

*<UML-Klassendiagramm der wesentlichen Klassen>*

*Klasse: DataHandler*

*Parameter:*

*Funktionen:*

* *setWohnort(plz : String, bundesland : String) : String*
* *setAdresse(pls : String, strasse : String, hausnummer : String, stadt : String) : String*
* *setAbteilung(kostenstelle : String, kuerzel : String, bezeichnung : String) : String*
* *setPerson(nName : String, vName : String, adress\_ID : String, Abteilung\_ID : String, eintrittd : String, geschlecht : String) : int*
* *deleteEntry(persNr : String) : int*
* *LoadData() : List<DataItem>*
* *FindEmployeeByPersonalNumber(personalNumber : String) : DataItem*
* *GetMySqlConnection() : String*
* *InitializeMySqlConnection() : MySqlConnection*
* *OpenNewPage(pageName : String, currentWindows : Window) : void*

*Klasse: DataItem*

*Parameter:*

* *+Personalnummer : int*
* *+Nachname : String*
* *+Vorname : String*
* *Eintrittsdatum : String*
* *Geschlecht : String*
* *Straße : String*
* *Hausnummer : int*
* *PLZ : int*
* *Stadt : String*
* *Bundesland : String*
* *Abteilungsbezeichnung : String*
* *Abteilungskuerzel : String*
* *Kostenstelle : int*

*Funktionen:*

*Klasse: Mitarbeiter\_Software.Export\_Final*

*Parameter:*

* *-mySqlConnection : MySqlConnection*
* *AllDataItems : List<DataItem>*

*Funktionen:*

* *+Export\_Final() : void*
* *-InitializeDatabaseConnection() : void*
* *-LoadData() : void*
* *-ExportButton\_Click(sender : object, e : RoutedEventArgs) : void*
* *-ExportToJSON(items List<DataItem>) : void*
* *-SelectAllButton\_Click(sender : object, e : RoutedEventArgs) : void*
* *-TxtSearch\_TextChanged(sender : object, e : TextChangedEventArgs) : void*
* *-FilterData(searchText : String) : void*
* *NavigateToNewPage\_Click(sender : object, e : RoutedEventArgs) : void*

*Klasse: Benutzeranlegen\_Final*

*Parameter:*

* *nName : String*
* *vName : String*
* *personalnr : String*
* *straße : String*
* *hausnr : String*
* *plz : String*
* *ort : String*
* *eintrittd : String*
* *geschlecht : String*
* *kostenstelle : String*
* *kuerzel : String*
* *bezeichnung : String*
* *bundesland : String*

*Funktionen:*

* *-clickCreateEmployee(sender : object, e : RoutedEventArgs) : void*
* *-clickSearchEmployee(sender : object, e : RoutedEventArgs) : void*
* *-clickDeleteEmployee(sender : object, e : RoutedEventArgs) : void*
* *-NavigateToNewPage\_Click(sender : object, e : RoutedEventArgs) : void*

# Test

*<Tabellarische Beschreibung der wesentlichen Testfälle und deren Durchführung. Zum Beispiel mit folgenden Spalten.: Testnummer- Testbeschreibung- Erwartetes Ergebnis - Bestanden>*

|  |  |  |  |
| --- | --- | --- | --- |
| *Testnummer* | *Testbeschreibung* | *Erwartetes Ergebnis* | *Bestanden* |
| *1* | *Mitarbeiter anlegen:* *Es werden alle Felder (bis auf die Personalnummer) im “Benutzer anlegen” Fenster ausgefüllt und der Button “Mitarbeiter anlegen” geklickt.* | *Mitarbeiter wird erfolgreich angelegt und ist auch in der Datenbank zu sehen. Dabei wird unter Umständen auch eine Adresse und weitere abhängige Einträge angelegt.* | *✅* |
| *2* | *Mitarbeiter suchen:*  *Es wird nur das Personalnummer Feld mit einer existierenden Personalnummer gefüllt. Anschließend wird der Button “Mitarbeiter suchen” geklickt.* | *Alle Felder werden automatisch mit den Informationen der Person gefüllt.* | *✅* |
| *3* | *Mitarbeiter löschen:* *Es wird nur das Personalnummer Feld mit einer existierenden Personalnummer gefüllt. Anschließend wird der Button “Mitarbeiter löschen” geklickt.* | *Der Eintrag des Mitarbeiters und damit verbundene Informationen werden gelöscht. Wenn es der einzige Mitarbeiter an genau dieser Adresse war, wird auch die Adresse aus der Datenbank gelöscht.* | *✅* |
| *4* | *Mitarbeiterliste anzeigen:* *Es wird der Button “Mitarbeiterliste anzeigen” geklickt.* | *Das aktuelle Fenster schließt sich und ein neues Fenster, welches alle Mitarbeiter als Liste anzeigt, erscheint.* | *✅* |
| *5* | *Mitarbeiter exportieren:*  *Es werden in der Mitarbeiterliste einige Mitarbeiter ausgewählt und anschließend der “Exportieren” Button geklickt.* | *Eine neue JSON-Datei wird im Testordner erstellt. In dieser Datei sind die ausgewählten Einträge erhalten.* | *✅* |
| *6* | *Alle Mitarbeiter exportieren:*  *Es wird in der Mitarbeiterliste der Button “Alle Auswählen” geklickt.* | *Eine neue JSON-Datei wird im Testordner erstellt. Alle Einträge aus der Liste sind vorhanden.* | *✅* |
| *7* | *Suchen in der Liste:*  *In der Mitarbeiterliste wird im Suchfeld “Suche nach Vor- und/oder Nachname” wird ein Name aus der Liste eingegeben.* | *Während des Eingebens sollten bereits die nicht zutreffenden Einträge weggefiltert werden.* | *✅* |
| *8* | *Zurück zum Anlegenfenster:*  *Hierbei wird im Fenster der Mitarbeiterliste unten der “zurück” Button gedrückt.* | *Das aktuelle Mitarbeiterlistenfenster schließt sich und das ursprüngliche Fenster zum Mitarbeiter anlegen öffnet sich.* | *✅* |