|  |  |
| --- | --- |
| **Projektplan ‚Umzug‘** | |
| Gruppenmitglieder: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Versionsnummer | Datum | Änderungen | Bearbeiter |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 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>

### Abgrenzungskriterien

*<Funktionalität die das System nicht besitzt>*

### 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>*

### Demonstrationsprototyp Packet Tracer

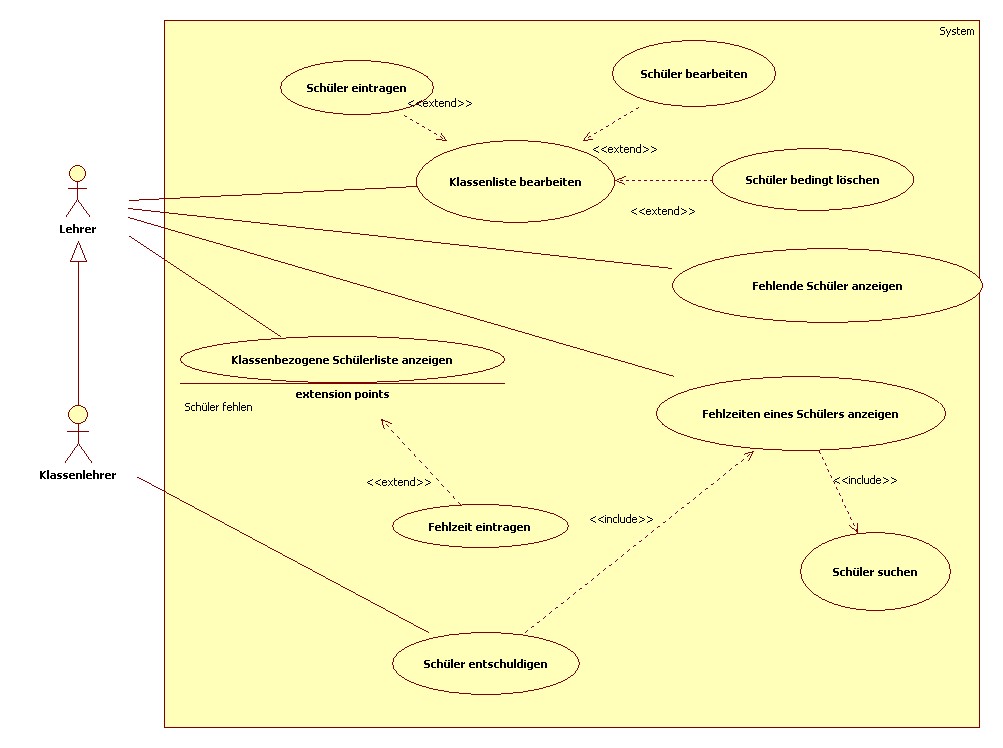
<Aufbau eines exemplarischen, funktionsfähigen Netzwerks>

# 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>*

|  |  |
| --- | --- |
| USE CASE: Schüler entschuldigen | |
| Primärer Aktor: | Klassenlehrer |
| Vorbedingung: | Fehlzeit vorhanden |
| Wichtigstes Erfolgsszenario: | 1. Lehrer sucht Schüler 2. Fehlzeiten des Schülers werden angezeigt 3. Fehlzeit wird bearbeitet 4. Fehlzeit wird gespeichert |
| Wichtige Varianten: | 2.a Fehlzeit befindet sich nicht im System obwohl Entschuldigung vorliegt  3.a Wechsel zu Use Case ‚Fehlzeit eintragen’ |
| 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: | 1. Lehrer wählt Klasse aus 2. Schülerliste wird angezeigt 3. Lehrer ändert Daten |
| Wichtige Varianten: | 2.a keine Schüler vorhanden  3.a Lehrer fügt Schüler hinzu |
| 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>*

# Test

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