

Projekt: VirtualCalendar

Domainanalyse

[Dokumentstruktur basiert auf RUP „Design Model“ und Studentenvorlage]

1. Dokumentinformationen

1.1. Änderungsgeschichte

<i>Datum</i>	<i>Version</i>	<i>Änderung</i>	<i>Autor</i>
20.04.2017	1	Erstellung	Günnewig
22.04.2017	2	Ergänzung	Draghiciu
23.04.2017	3	Ergänzung	Kuczera

1.2. Inhalt

1. Dokumentinformationen	2
1.1. Änderungsgeschichte.....	2
1.2. Inhalt.....	3
2. Einführung (Introduction).....	4
2.1. Übersicht	4
3. Domain Modell	4
3.1. Strukturdiagramm.....	4
3.2. Beschreibung der Domänenklassen (Konzeptionellen Klassen)	4
3.2.1. Domänenklasse 1: Benutzer.....	4
3.2.1.1. Beschreibung.....	4
3.2.1.2. Attribute	4
3.2.1.3. Beziehungen.....	5
3.2.2. Domänenklasse 2 : Admin.....	5
3.2.2.1. Beschreibung.....	5
3.2.2.2. Attribute	5
3.2.2.3. Beziehungen.....	5
3.2.3. Domänenklasse 3 : Gruppe	6
3.2.3.1. Beschreibung.....	6
3.2.3.2. Attribute	6
private List<Termin> List_Termine	6
3.2.3.3. Beziehungen.....	6
3.2.4. Domänenklasse 4 : Termin.....	6
3.2.4.1. Beschreibung.....	6
3.2.4.2. Attribute	6
3.2.4.3. Beziehungen.....	7
4. Systemsequenzdiagramme.....	7
4.1. Use Case Profil erstellen.....	7
4.2. Use Case Gruppe erstellen.....	8
4.3. Use Case Ansicht wählen	8
4.4. Use Case Termin erstellen	9
5. Systemoperationen	9
5.1.1. Vertrag Systemoperation 1	9
5.1.2. Vertrag Systemoperation 2	9
5.1.3. Vertrag Systemoperation 3	9
5.1.4. Vertrag Systemoperation 4	10

2. Einführung (Introduction)

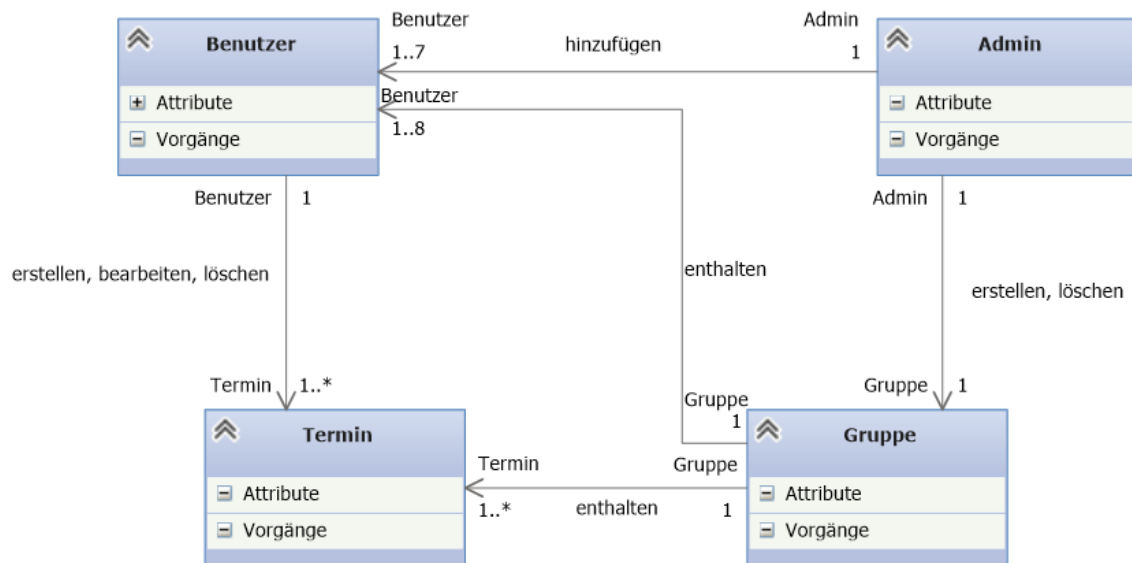
2.1. Übersicht

Im nachfolgenden Teil des Dokumentes wird das Domainmodell näher erläutert, sowie die Sequenzdiagramme der wichtigsten Use-cases.

3. Domain Modell

3.1. Strukturdiagramm

cd UMLClassDiagram1



3.2. Beschreibung der Domänenklassen (Konzeptionellen Klassen)

3.2.1. Domänenklasse 1: Benutzer

3.2.1.1. Beschreibung

Die Klasse Benutzer ist die Klasse wo die Attribute der User und ihre Funktionalitäten festgelegt werden.

Der Benutzer kann sich ein Profil anlegen und danach meldet er sich in einer Gruppe an.

Er kann zwischen Einzel- und Gruppenansicht wählen und dort Termine erstellen.

Falls sich ein Termin ändert kann er diesen bearbeiten oder löschen.

Ein Benutzer wird von einem Admin hinzugefügt.

3.2.1.2. Attribute

Von jedem Benutzer werden folgende Daten gespeichert: Name, eine Benutzer-ID, das Geburtsdatum, die E-Mail Adresse, ein persönliches Passwort und ein Avatar (Profilbild).

```
public string Str_Name
public int Id
public DateTime Dt_GeburtsDatum
public string Str_Email
public string Str_Passwort
public string Str_Avatar
```

3.2.1.3. Beziehungen

Die Klasse Benutzer hat eine Beziehung zu der Klasse Termin, da dieser Termine erstellt, bearbeitet und löscht.

Zudem hat die Klasse Benutzer eine Beziehung zur Klasse Gruppe, da er ein Teil der Gruppe ist.

Die Klasse Benutzer hat auf jedenfall eine Beziehung zu der Klasse Admin, da ein Benutzer von einem Admin in die Gruppe hinzugefügt wird.

3.2.2. Domänenklasse 2 : Admin

3.2.2.1. Beschreibung

Ein Admin erbt von der Klasse Benutzer. Er hat zusätzlich die Möglichkeit, eine Gruppe zu erstellen (bei diesem Vorgang legt der Admin einen Gruppennamen sowie ein Gruppenpasswort fest). Der Admin kann bis zu sieben Benutzer zu dieser Gruppe hinzuzufügen, in dem er ihre E-Mail Adressen angibt. Er kann auch vorhandene Benutzer aus der Gruppe entfernen, oder die Gruppe löschen.

3.2.2.2. Attribute

Alle Attribute werden von der Klasse Benutzer geerbt:

```
public string Str_Name
public int Id
public DateTime Dt_GeburtsDatum :
public string Str_Email
public string Str_Passwort
public string Str_Avatar
```

3.2.2.3. Beziehungen

Die Klasse Admin erbt alle Attribute und Funktionen von der Klasse Benutzer und hat zusätzlich vier eigene Funktionen.

Die Klasse Admin hat eine Beziehung zur Klasse Gruppe, da er eine Gruppe erstellen und falls gewünscht wieder löschen kann.

Die Klasse Admin hat eine Vererbungsbeziehung zur Klasse Benutzer, da er sich von einem Benutzer nur durch vier weitere Funktionen unterscheidet:

```
private void gruppeErstellen() { }
private void gruppeLoeschen() { }
private void benutzerHinzufuegen() { }
private void benutzerEntfernen() { }
```

Eine Beziehung zur Klasse Benutzer besteht auch, da der Admin zur Gruppe Benutzer hinzufügen kann.

3.2.3. Domänenklasse 3 : Gruppe

3.2.3.1. Beschreibung

Eine Gruppe besteht aus einem bis acht Benutzern (einer von ihnen ist der Admin, der in der Gruppe sich wie ein normaler Benutzer verhält). In der Gruppenansicht kann man die gemeinsamen Termine aufeinander abstimmen und einsehen.

3.2.3.2. Attribute

Jede Gruppe verfügt über folgende Attribute: einen Gruppennamen, ein Gruppenpasswort, eine maximale Benutzeranzahl (wird von uns mit acht initialisiert), eine Liste der Benutzer und eine Liste der Termine, die von den Benutzern in der Gruppe eingetragen werden.

```
public string Str_GruppenName
public string Str_Passwort
public int Max_benutzeranzahl
public List<Benutzer> List_Benutzer
private List<Termin> List_Termine
```

3.2.3.3. Beziehungen

Die Klasse Gruppe steht in der Beziehung mit der Klasse Admin, da der Admin die Gruppe erstellt sowie löscht.

Außerdem steht die Klasse Gruppe in der Beziehung mit der Klasse Benutzer, da die Benutzer Mitglieder der Gruppe sind.

Zudem steht die Klasse Gruppe in der Beziehung mit der Klasse Termin, da es Termine für die ganze Gruppe gibt.

3.2.4. Domänenklasse 4 : Termin

3.2.4.1. Beschreibung

Termine sind die Einträge im Kalender, die in den Ansichten angezeigt werden.

Für den Termin wird ein Datum und eine Zeitspanne festgelegt, dieser wird durch einen Beginn und ein Ende gekennzeichnet. Eine individuelle Wiederholung ist möglich.

3.2.4.2. Attribute

Jeder Termin besitzt folgende Attribute:

die Kalenderansicht, in der er gespeichert/ gezeigt werden soll, eine Bezeichnung (z.B. „Arzttermin“), einen Beginn und ein Ende (ausgedrückt durch Datum und Uhrzeit), den Namen und die ID des Benutzers, der den Termin anlegt, eine Terminbeschreibung (z.B. der Name oder die Adresse des Arztes, bei dem der Benutzer den Termin hat), einen Wiederholungszyklus, da der Benutzer die Möglichkeit hat, einen sich wiederholenden Termin (z.B. ein Geburtstag) anzulegen, und eine Termin-ID (eine fortlaufende Zahl der Termine).

```
public enum Enum_KalenderAuswahl { Gruppenansicht, Individualansicht }
public string Str_Bezeichnung
public DateTime Dt_Beginn
public DateTime Dt_Ende
public Benutzer Benutzer
public int BenutzerId
public string Str_TerminBeschreibung
public enum Enum_WiederholungsZyklus { jaehrlich, monatlich, woeentlich,
taeglich, keine }
```

```
public int Id
```

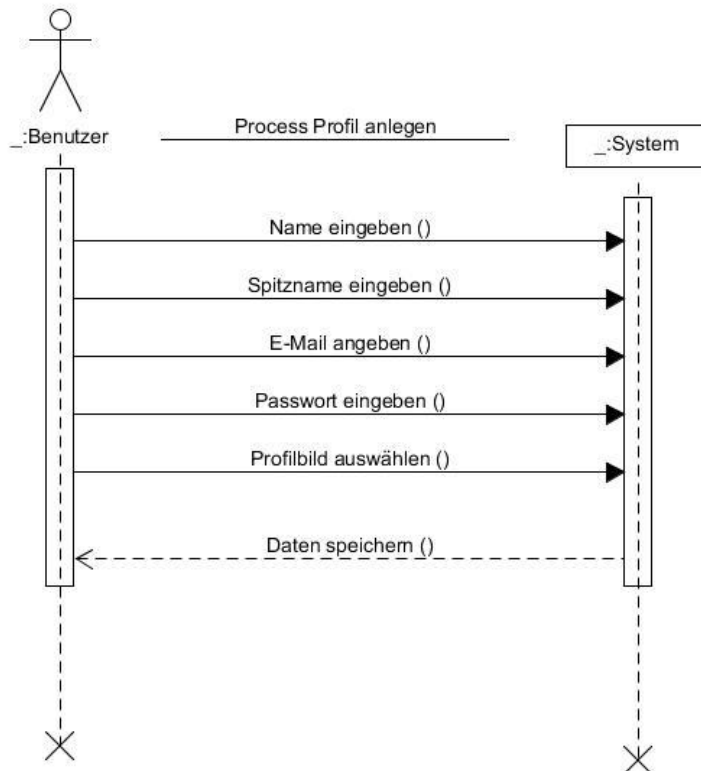
3.2.4.3. Beziehungen

Die Klasse Termin steht in der Beziehung mit der Klasse Benutzer, da der Benutzer Zugriff auf die Termine hat.

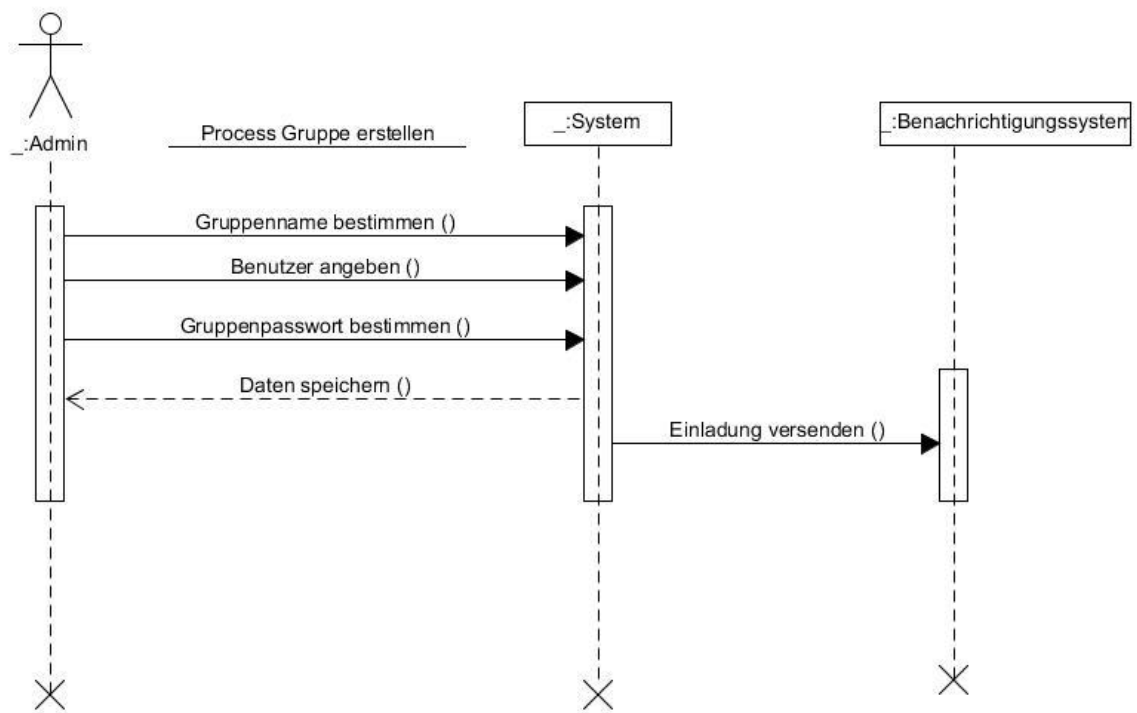
Die Klasse Termin hat ebenfalls eine Beziehung zu der Klasse Gruppe, da die Termine in der Gruppe enthalten sind.

4. Systemsequenzdiagramme

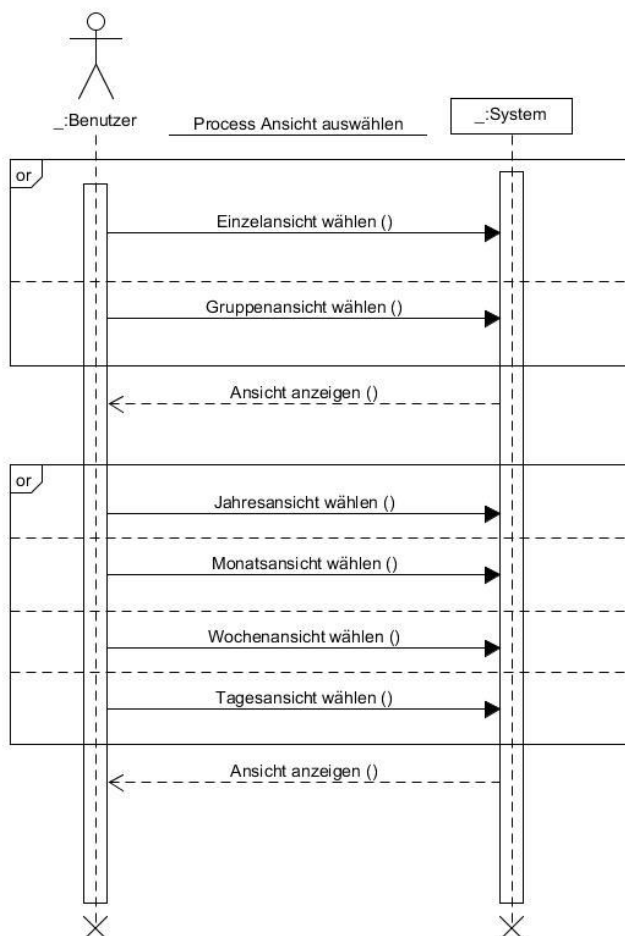
4.1. Use Case Profil erstellen



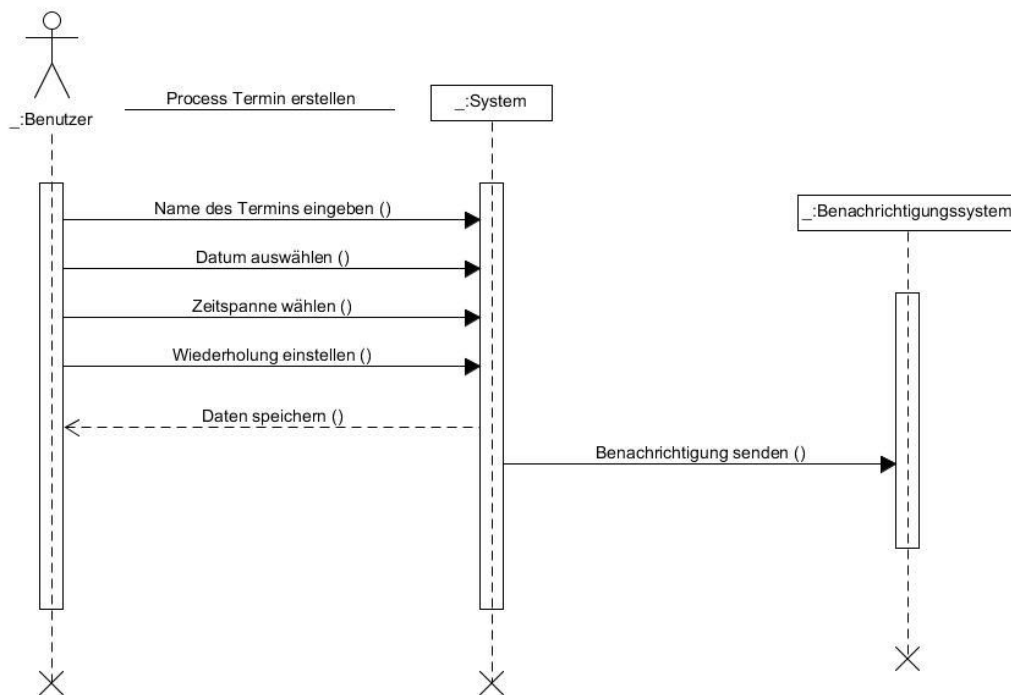
4.2. Use Case Gruppe erstellen



4.3. Use Case Ansicht wählen



4.4. Use Case Termin erstellen



5. Systemoperationen

<Beschreibung der Systemoperationen aus den SSDs>

5.1.1. Vertrag Systemoperation 1

Process Profil anlegen:

Mit diesem Prozess soll der Nutzer ein Profil anlegen mit Name, E-Mail und Passwort, damit niemand anderes bei der Anmeldung auf dieses Konto zugreifen kann außer diesem Nutzer. Der Benutzer kann sich ebenfalls einen Spitznamen, sowie ein gegebenes Bild als Profilbild zulegen, an dem ihn die anderen Benutzer erkennen. Anschließend werden die Daten des neuen Profils vom System eingespeichert, so dass die zukünftige Nutzung gewährt ist. Die Vorbedingung damit der Benutzer sich ein Profil anlegen kann, ist dass die Software installiert ist. Wenn das Profil angelegt ist, dann hat der Benutzer erst die Möglichkeit auf den individuellen Kalender zugreifen zu können. Damit allerdings der Kalender als Gruppe verwendet werden kann muss erst eine Gruppe erstellt werden.

5.1.2. Vertrag Systemoperation 2

Process Gruppe erstellen:

Mit diesem Prozess kann der Administrator eine Gruppe erstellen, indem er als erstes einen Gruppen Namen bestimmt. Anschließend gibt der Administrator die anderen Benutzer an, welche mit ihm den VirtualCalendar nutzen sollen. Damit keine anderen Nutzer auf diese Gruppe zugreifen können, wird vom Administrator noch ein Gruppenpasswort bestimmt. Danach speichert das System die Gruppe und sendet an alle angegebenen Nutzer per E-Mail eine Einladung, wo der Gruppenname und das Gruppenpasswort angegeben ist. Die Vorbedingung zur Gruppen Erstellung ist, dass der Administrator sich bereits ein Profil angelegt hat. Durch die Erstellung der Gruppe, entsteht eine Ansicht für einen Gruppenkalender, welche von allen Benutzern verwendet werden kann die sich in diese Gruppe angemeldet haben. Die Nachbedingung dass der VirtualCalendar seinen Zweck erfüllt ist, dass die Nutzer auch Termine in diesen Kalender eingeben.

5.1.3. Vertrag Systemoperation 3

Process Ansicht auswählen:

Die Benutzer haben die Möglichkeit zwischen Einzelansicht und Gruppenansicht zu wählen, welches anschließend vom System angezeigt wird. Danach können Sie noch zwischen Jahres-, Monats-, Wochen- und Tagesansicht wählen. Danach zeigt das System die ausgewählte Ansicht an. Somit wird es ermöglicht, dass der Benutzer die bereits eingetragenen Termine sieht oder noch Termine eingeben kann. Die Vorbedingung ist, dass man sich angemeldet hat, sowohl in der Profil Anmeldung als auch in der Gruppen Anmeldung.

5.1.4. Vertrag Systemoperation 4

Process Termin erstellen:

Damit ein Termin eindeutig zugeordnet werden kann, muss der Benutzer bei der Terminerstellung als Erstes einen Namen für den Termin eintragen und zusätzlich das gewünschte Datum, an dem das Ereignis stattfinden soll. Da auch Termine über mehrere Stunden oder Tage stattfinden können, ist auch immer eine Zeitspanne einzugeben. Zuletzt hat der Benutzer die Möglichkeit das Ereignis sich Wiederholen zu lassen. Zum Beispiel wöchentlich (wie bei einem Meeting) oder jährlich (wie bei einem Geburtstag). Dieser Termin wird dann vom System im Kalender eingespeichert. Damit die einzelnen Benutzer eine Kalender Veränderung mitbekommen, bekommen sie durch das Benachrichtigungssystem eine Nachricht nach der Terminerstellung. Die Vorbedingung zur Terminerstellung ist, dass die beiden Anmeldungen erfolgt sind.