Projekt: VirtualCalendar

Software Architektur Spezifikation (Software Architecture Document)

[Dokumentstruktur basiert auf RUP „Software Architecture Document“]

1.Dokumentinformationen

1.1 Änderungsgeschichte

Datum	*Version*	*Änderung*	*Autor*
24.04.2017	1.0	Erstellung	Kuczera
04.05.2017	2.0	Bearbeitung	Kuczera
10.05.2017	3.0	Bearbeitung	Kuczera
11.05.2017	4.0	Bearbeitung und Fertigstellung	Draghiciu
18.05.2017	5.0	Bearbeitung nach Review	Artinger

1.2.Inhalt

2 Einführung (Introduction)

2.2 Referenzen (References)

- https://de.wikipedia.org/wiki/Grafische_Benutzeroberfl%C3%A4che

2.3 Übersicht (Overview)

Im nachfolgenden Dokument werden die Architektur und derer Komponenten beschrieben. Dort wird genauer erläutert wieso wir uns für diese entschieden haben, danach verdeutlichen wir diese mit verschiedenen Diagrammen. Wir erklären was Threads sind und wie sie im Zusammenhang mit unserer Software fungieren. Am Ende gehen wir noch auf die Datenspeicherung, Größen und Leistungen ein, die für unser Programm erforderlich sind.

3 Architektonische Darstellung (Architectural Representation)

Wir haben uns für die Schichten-Architektur entschieden, da unsere Komponenten voneinander abhängen. Diese Architektur schafft uns den Vorteil, dass die Schichten logisch voneinander getrennt sind. Unsere Schichtenarchitektur ist aufgeteilt in GUI, Domänenschicht und E-Mail-Benachrichtigungssystem.
Das E-Mail-Benachrichtigungssystem informiert die Benutzer, welche Termine zu welchem Zeitpunkt stattfinden und ob Terminänderungen vorgenommen worden sind.
Die Klassen können nur in vollem Umfang funktionieren, wenn das Gruppenpasswort und der Gruppenname durch das E-Mail-Benachrichtigungssystem weitergegeben werden.
Durch die GUI wird die Schnittstelle zum Benutzer hergestellt. Die GUI beinhaltet verschiedene Kalenderansichten.
Ein externer online Server erlaubt, dass jeder Benutzer durch die Benutzeroberfläche jederzeit Daten aufrufen oder sich anmelden kann.

4 Architektonische Ziele & Einschränkungen (Architectural Goals and Constraints)

Mit dieser Architektur kann die Abhängigkeit unter den Komponenten unseres Programmes reduziert werden. Dies erleichtert die Wartbarkeit, da man die einzelnen Komponenten bearbeiten kann, ohne dass das gesamte System verändert wird.
Die Abhängigkeit der Komponenten unseres Programmes voneinander stellt seine Schwachstelle dar.
Zum einen ist die Verknüpfung mit der Datenbank von sehr großer Bedeutung, denn ohne sie würden die Benutzerprofile und ihre Termine nicht eingespeichert werden. Das bedeutet, dass ohne die Datenbank unsere Software ihre Aufgabe nicht erfüllen würde.
Zum anderen kann für unser Programm das E-Mail-Benachrichtigungssystem zum Problem werden, da wir die Einladung für unsere Software mit dem entsprechenden Gruppennamen und Gruppenpasswort per E-Mail versenden. Genauso sollen Benachrichtigungen durch das E-Mail-Benachrichtigungssystem versendet werden, wenn Termine erstellt, gelöscht oder geändert werden. Der Ausfall des E-Mailbenachrichtigungssystems würde bedeuten, dass man den Gruppennamen, sowie das Gruppenpasswort mündlich weitergeben müsste und dass der Benutzer über Änderungen im Kalender nicht informiert werden würde.
GUI ist für unsere Benutzerschnittstelle wichtig, denn sie ist die visuelle Darstellung in unserem Programm. Wenn diese nicht funktionieren würde, könnte man die Kalender und die dazugehörigen Termine nicht sehen. Außerdem könnte der Benutzer die Software nicht verwenden, da er z.B. keine Buttons zur Termin-, Profil- und Gruppenerstellung sehen würde.

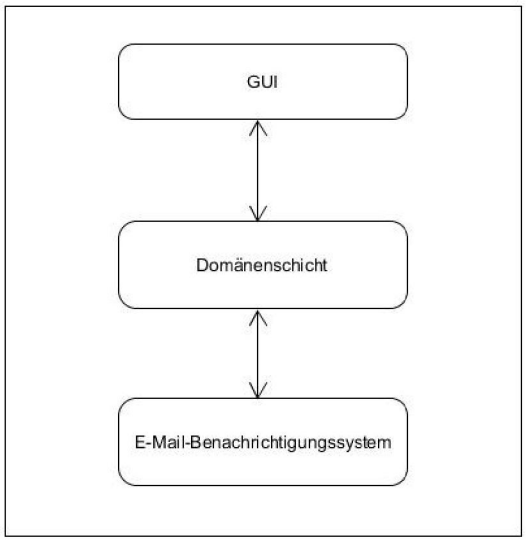
Wie bereits erwähnt, müssen auf den Computern, sowie auf den Handys Windows10 installiert und eine Verbindung zum Internet vorhanden sein, damit unser Programm funktioniert und die Datenverbindung hergestellt werden kann.

5 logische Architektur (Logical View)

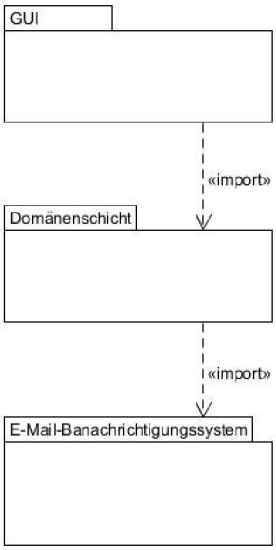
Wie bereits erwähnt besteht der VirtualCalendar aus drei Schichten bzw. Packages: die GUI, die Problem-Domain und das E-Mail-Benachrichtigungssystem.

5.1 Übersicht (Overview)

Schichten-Architektur:



Übersichtsdiagramm:



5.2 Design Pakete (Architecturally Significant Design Packages)

5.2.1 Package GUI

5.2.1.1 Beschreibung des Package

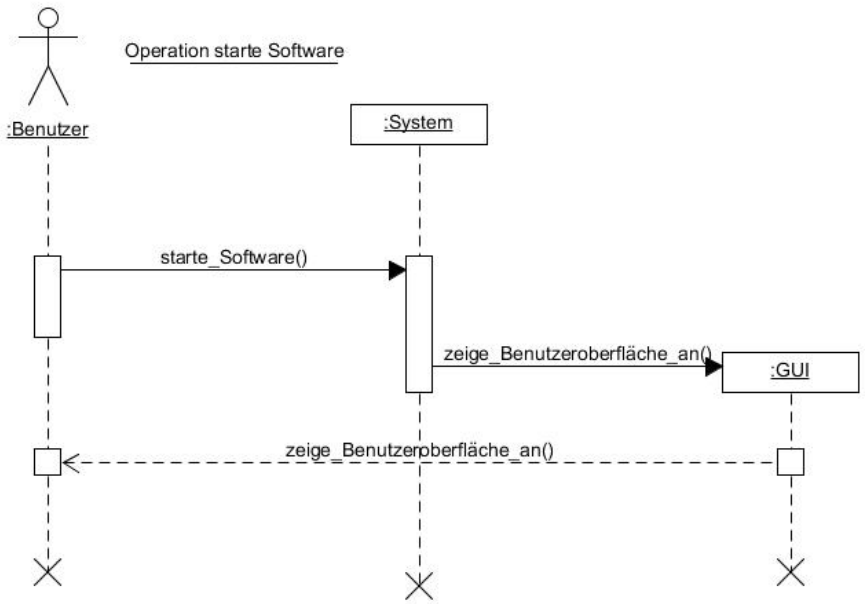
GUI ist die Schnittstelle zum Benutzer. Sie hat die Aufgabe die Software mit graphischen Symbolen, Steuerelementen und Widgets auszustatten und bedienbar zu machen. Dies würde für unsere Software bedeuten, dass man durch die GUI-Gestaltung zwischen den einzelnen Seiten navigieren kann. Von der Startseite gelangt man zu den verschiedenen Kalenderansichten, zu einer Softwareinformationsseite und zur Profilbearbeitungsseite. Wenn man einen Termin erstellen möchte, öffnet sich eine neue Seite mit auszufüllenden Felder, die die Details des Termins beinhalten, wie z.B. Beschreibung, Datum, Uhrzeit, etc. Die Möglichkeit zur Terminerstellung hat man auf jeder Kalenderansichtsseite, außer auf der Jahresansichtsseite.

5.2.1.2 Schnittstellen

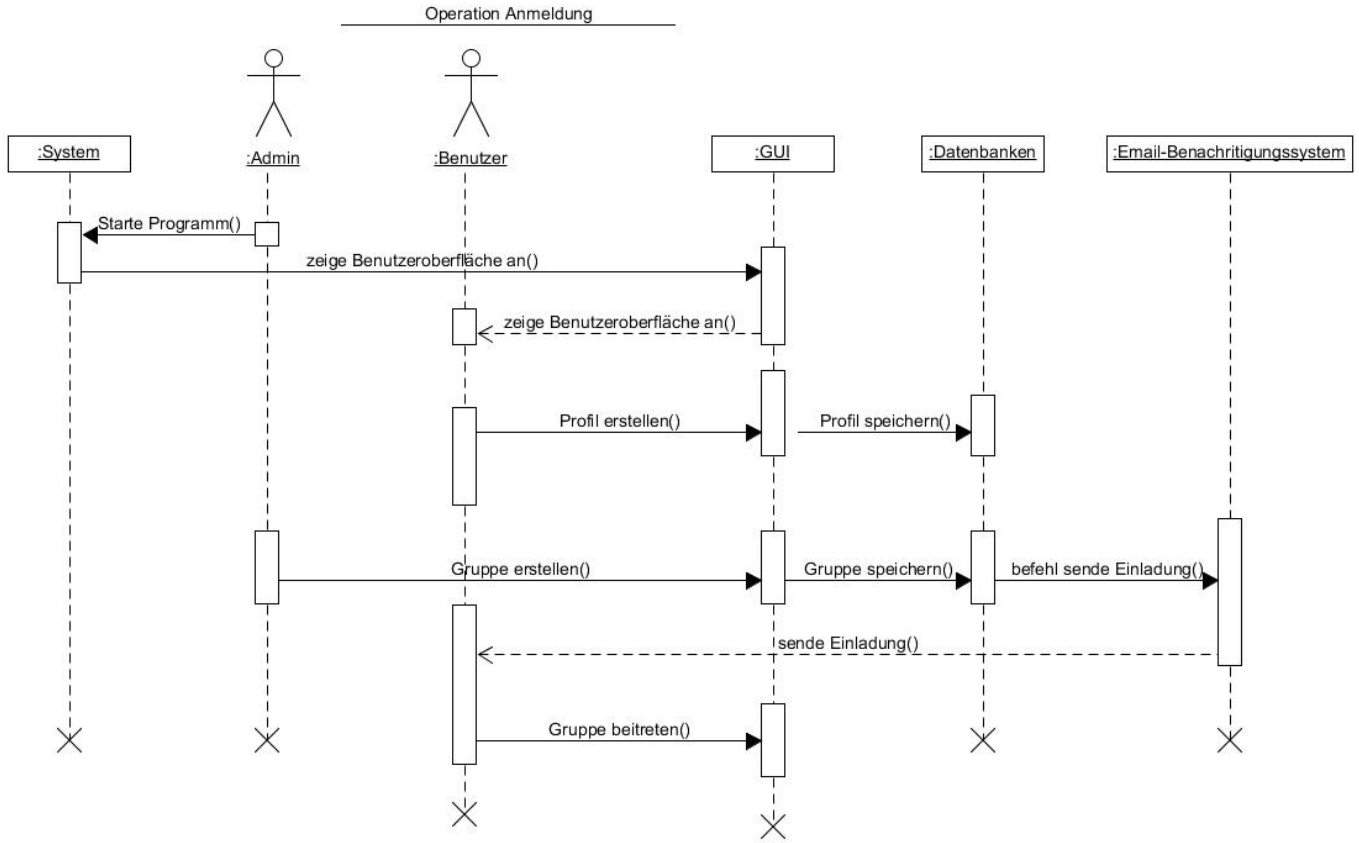
Der Benutzer öffnet die Software und sieht das Hauptfenster (Startseite).
Bei der ersten Nutzung des Programmes muss ein Benutzer ein Profil anlegen. Dies kann er machen, indem er den Button «Profil erstellen» anklickt. Somit gelangt er zu einer Anmeldeseite, wo er seine eigene E-Mail-Adresse und seine Profildaten eingibt. Die Daten werden in der Datenbank gespeichert.
Ebenfalls auf der Startseite gibt es auch den Button «Gruppe anmelden», wo ein Benutzer die Admin-Rolle übernimmt und andere Benutzer zu einer gemeinsamen Gruppe einladen kann, indem er ihre E-Mail-Adressen, einen Gruppennamen und -passwort eingibt. An die angegebenen E-Mail-Adressen werden Einladungen zum Beitreten der Gruppe geschickt. Diese beinhalten den Gruppennamen und das gemeinsame Passwort der Gruppe. Diese muss der Benutzer eingeben, indem er auf den Button «Gruppe beitreten» klickt. Dieser wird ebenfalls auf der Startseite angezeigt. Diese Daten werden ebenfalls in der Datenbank gespeichert.
Von der Startseite aus hat der Benutzer dann die Möglichkeit durch einen SplitView-Button in die einzelnen Ansichten zu wechseln (Buttons zu jeder Ansicht sind immer am linken Rand vorhanden , aber durch den SplitView-Button wird der Name angezeigt, z.B. „Tagesansicht“).
Der erste Button ist der Startseite-Button. Damit kann der Nutzer wieder auf die Startseite gelangen, wenn er auf einer anderen Seite ist.
Es gibt vier weitere verschiedene Buttons für die Kalenderansichten, nämlich die Tages-, Wochen- Monats- und Jahresansicht. Durch das Anklicken dieser Buttons kommt man auf die gewünschten Seiten.
Auf jeder Kalenderansichtsseite gibt es einen Button, durch den der Benutzer zwischen Einzel- und Gesamtansicht wählen kann. In der Einzelansicht können nur die eigenen Termine in der ausgesuchten Kalenderansicht angezeigt werden. In der Gesamtansicht werden die Termine der gesamten Gruppe in der ausgewählten Kalenderansicht angezeigt.
Unter den vier Ansichtsbuttons gibt es einen Profilbearbeitungsbutton, wo der Benutzer sein Profil bearbeiten kann.
Außerdem gibt es noch einen Button für Informationen. Dieser führt zu einer Seite, wo Informationen zu der Software zu sehen sind.

5.2.1.3 Operationen

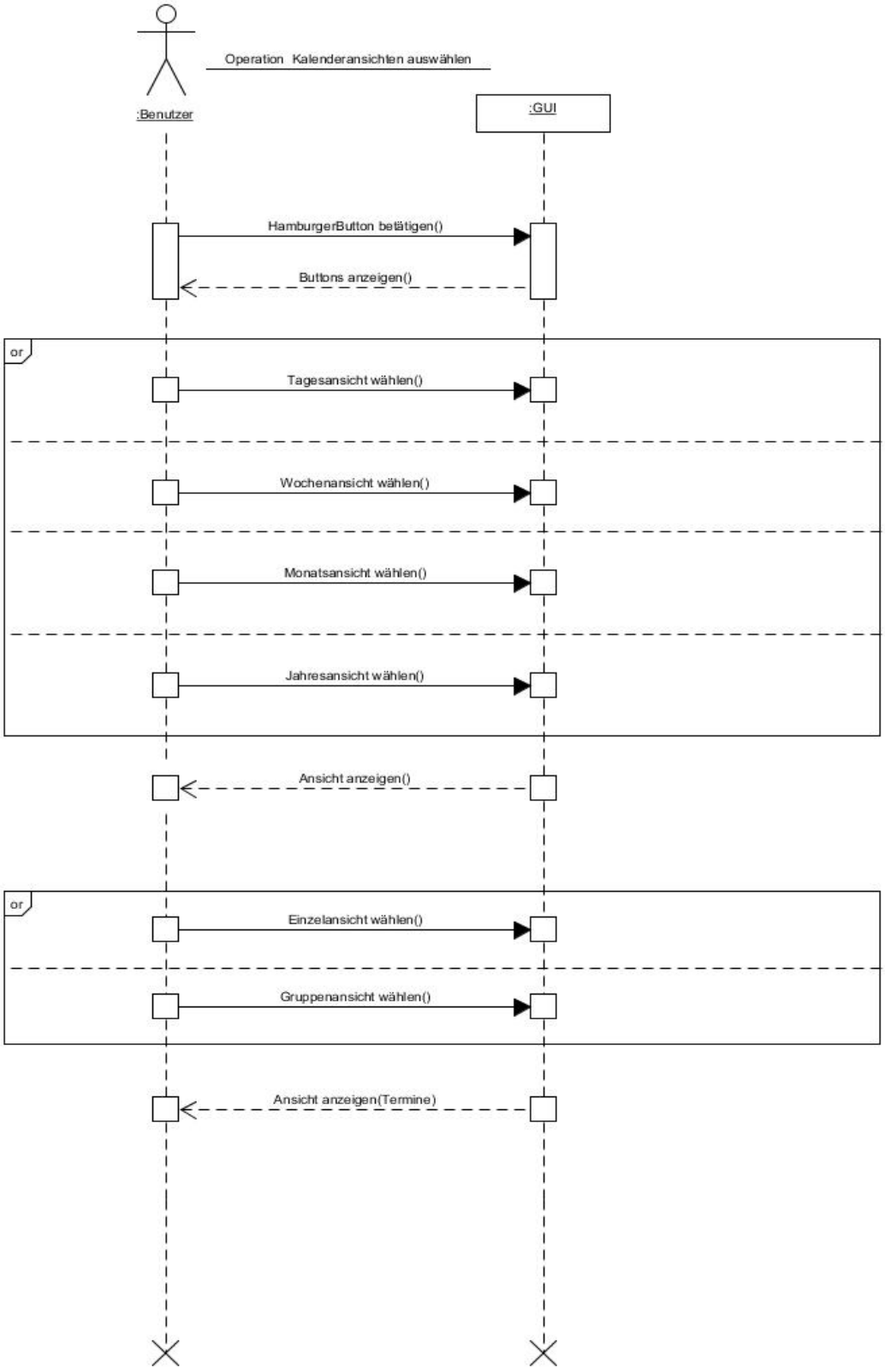
5.2.1.3.1 ‚Starte Software‘



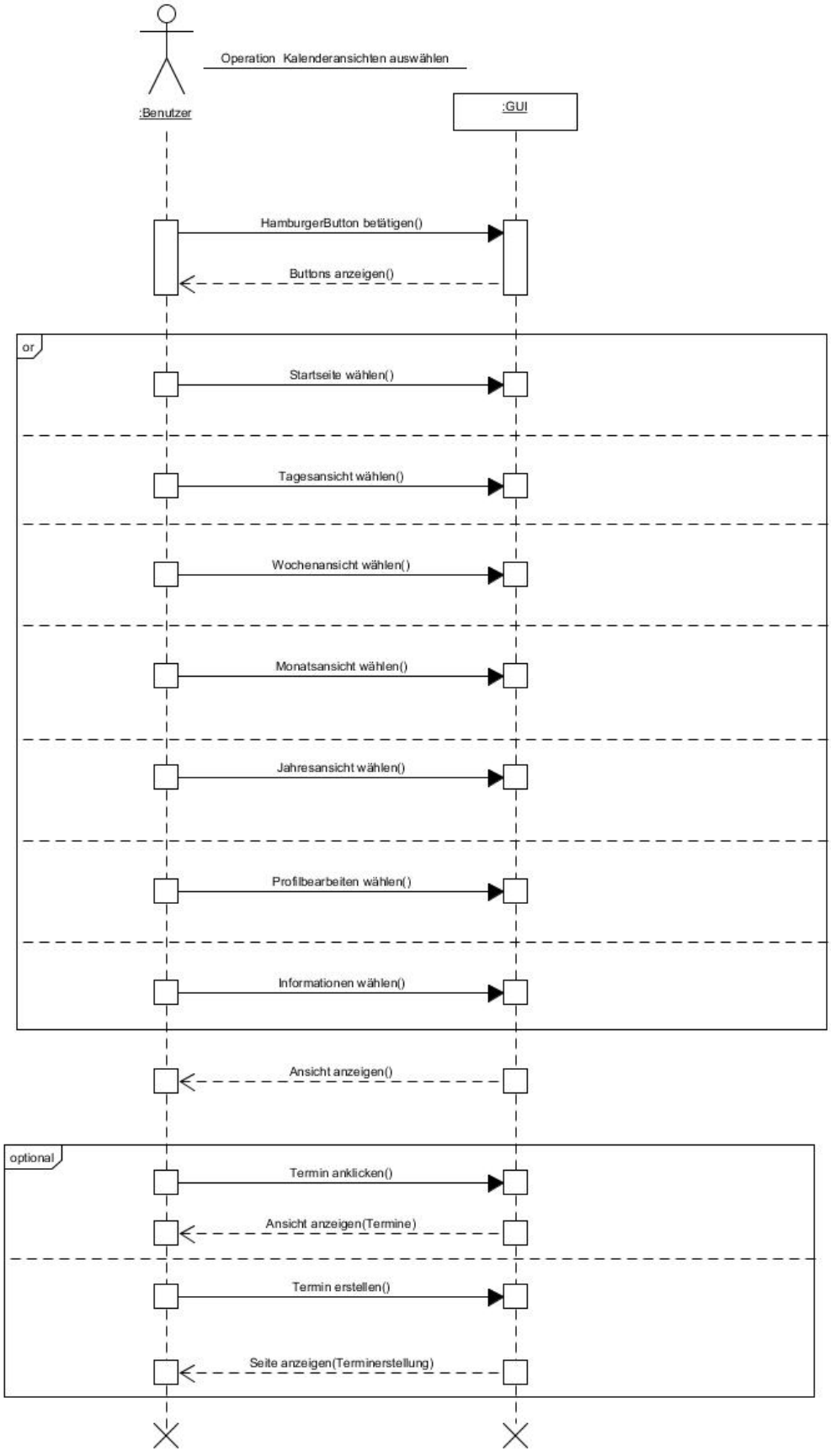
5.2.1.3.2 ‚Anmelden‘



5.2.1.3.3 ‚Kalenderansicht wählen‘



5.2.1.3.4 ‚Seiten wechseln‘



5.2.2 Package Problem-Domain

5.2.2.1 Beschreibung des Package

Das neue Fachgebiet beschäftigt uns mit den Kalenderanwendungen bezüglich
der Jahres-, Monats-, Wochen- und Tagesansichten sowie auch das Bedenken der Schaltjahre und der unterschiedlichen Anzahl von Monatstagen.
Ein weiterer Teil der Problem-Domain ist die Termingestaltung. In diesem Gebiet beschäftigen wir uns mit der Übersichtlichkeit der Termine, die vor allem dann sehr wichtig ist, wenn mehrere Personen am selben Tag verschiedene oder gleiche Termine haben.

5.2.2.2 Schnittstellen

Die Schnittstelle zur Problem-Domain ist vor allem die Gestaltung der verschiedenen Kalenderansichten und ihre Programmierung, so dass alle Ausnahmen wie z.B. Schaltjahre berücksichtigt werden. Durch mathematische Formeln und Algorithmen werden die Schaltjahre bestimmt und die Tagesanzahl der Monate entsprechend angepasst.
Um die Übersichtlichkeit der Termine zu garantieren, wird die Wochenansicht so programmiert, dass die Tabelle sich dynamisch anpasst, je nachdem wie viele Termine eingetragen sind. Bei der Monatsansicht werden nur die Farben der Benutzer in der Zelle des angezeigt, die einen Termin an dem Tag haben. Außerdem kann man in der Monatsansicht Geburtstage direkt sehen, indem ein Symbol in der Tageszelle angezeigt wird.

5.2.3 Package Datenbanksystem

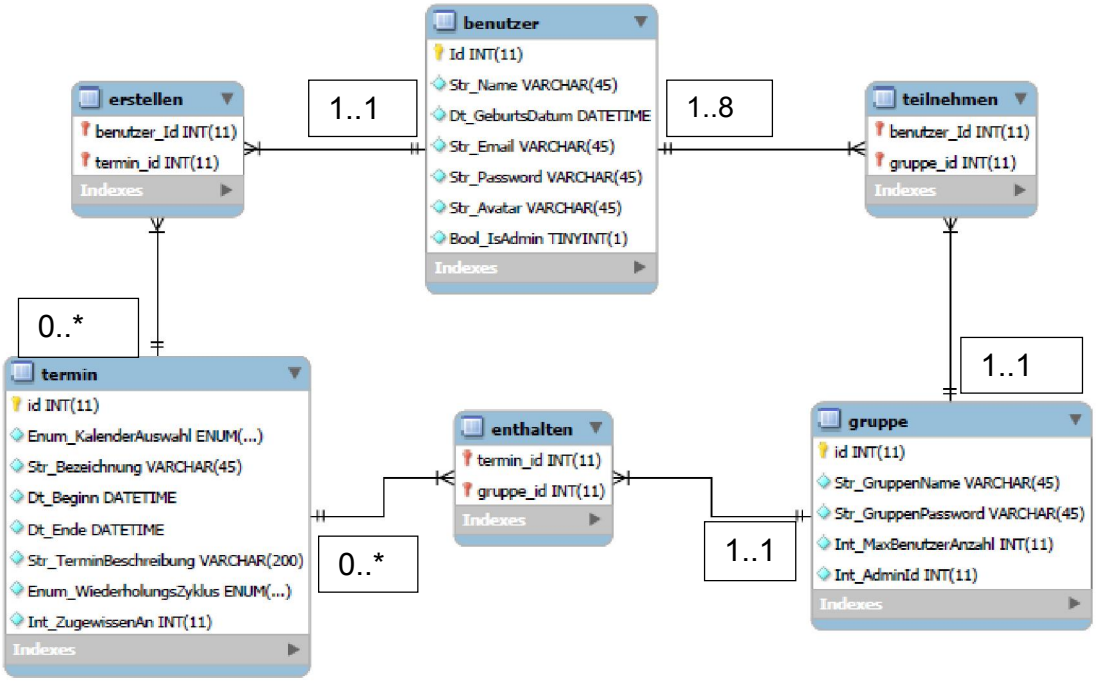
5.2.3.1 Beschreibung des Package

Die Datenbank beinhaltet drei Tabellen, in denen die Daten der Benutzer, der Termine und der Gruppe gespeichert werden.
Eine Gruppe kann einen bis acht Benutzer beinhalten, wobei ein Benutzer nur zu einer einzigen Gruppe gehören darf. Eine Gruppe kann beliebig viele Termine beinhalten. Die Benutzer dürfen ebenfalls beliebig viele Termine erstellen.

5.2.3.2 Schnittstellen

Die Verbindung zwischen der Datenbank und dem Programm erfolgt über einen online Server (Virtuelle Maschine). Die SQL-Befehle werden im Programmcode durch Funktionen definiert. Dadurch werden die Daten aus der Datenbank aufgerufen, geändert oder gelöscht. Die gute Funktionalität der Datenbankverbindung ist für unsere Software von sehr großer Bedeutung.

5.2.3.3 Diagramme



5.2.4 Package Email-Benachrichtigungssystem

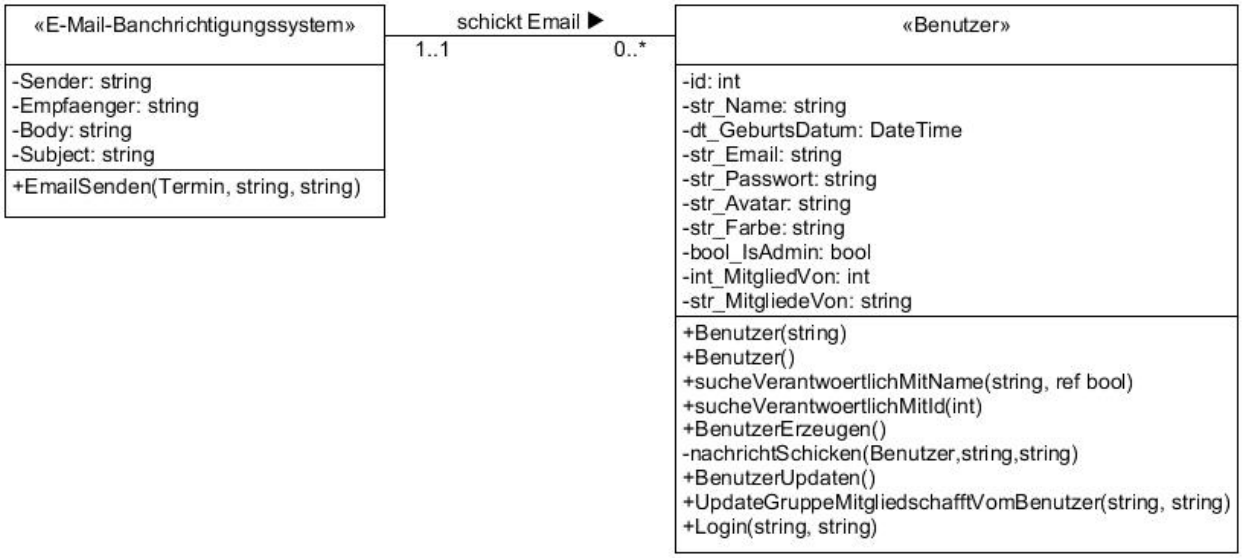
5.2.4.1 Beschreibung des Package

Das E-Mail-Benachrichtigungssystem wird gebraucht, um Erinnerungen und Einladungen zu verschicken. Die Benutzer können für ihre Termine Erinnerungen einstellen und auch Wiederholungen für die Erinnerungen, die dann über E-Mail empfangen werden. Außerdem wird die Einladung zu einer Gruppe über E-Mail versendet, welche auch den Gruppennamen und –passwort beinhaltet.

5.2.4.2 Schnittstellen

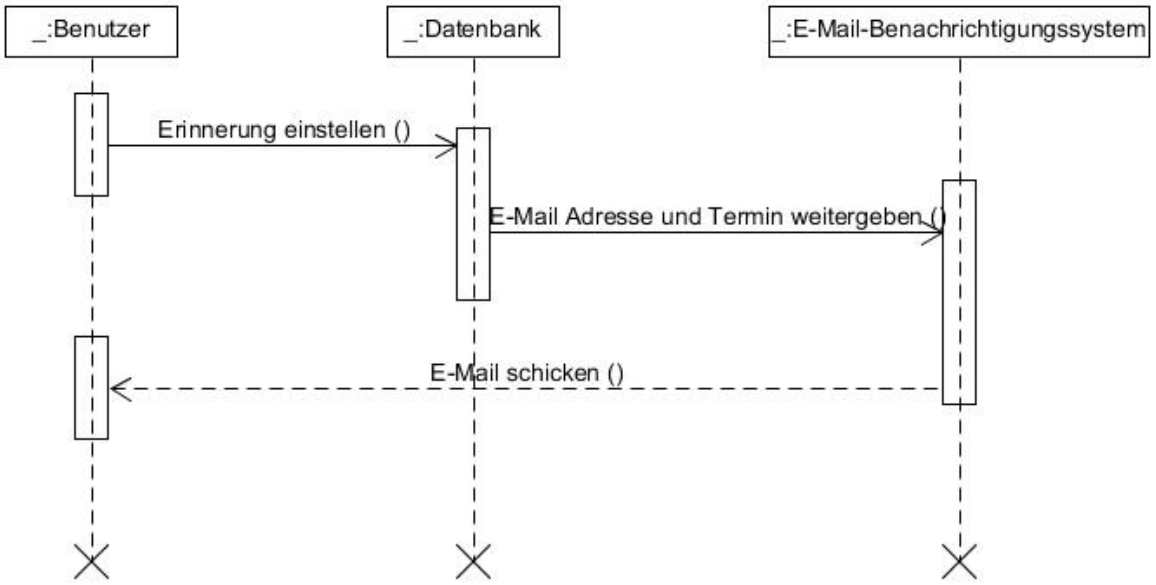
Das E-Mail-Benachrichtigungssystem braucht eine Schnittstelle zu unserem Programm und zu der Datenbank, da dort die E-Mail Adressen der Nutzer gespeichert werden. Die Schnittstelle zu unserem Programm ist Outlook. Durch den Outlook Client werden E-Mails mit unserer E-Mail Adresse an die Nutzer versendet.

5.2.4.3 Diagramme

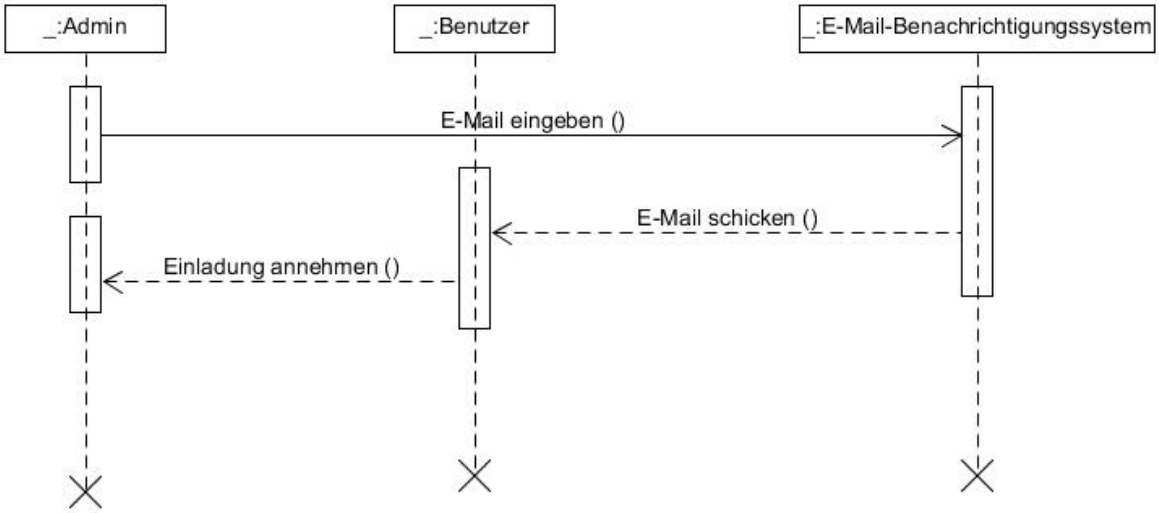


5.2.4.4 Operationen

5.2.4.4.1 ‚Erinnerung schicken'

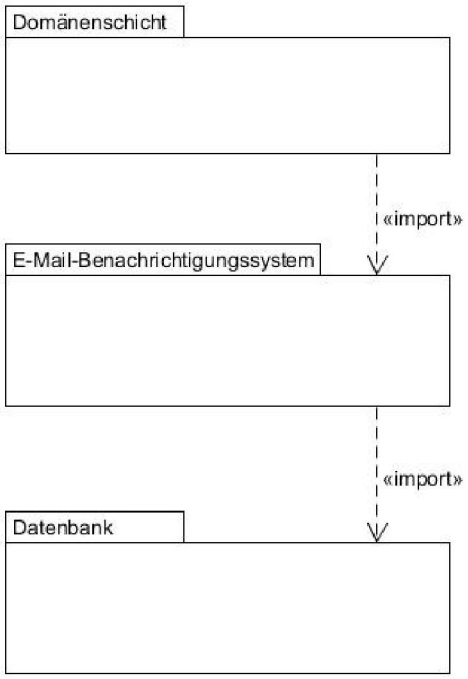


5.2.4.4.2 ‚Einladung schicken'



6 Physikalische Sicht (Physical View)

Die Datenbank läuft auf der virtuellen Maschine und ist mit unserem Programm über einen Server verbunden.
Das Programm selbst läuft auf jedem Rechner, auf welchem das Programm installiert wird.
Das E-Mail-Benachrichtigungssystem läuft auch über einen Server.



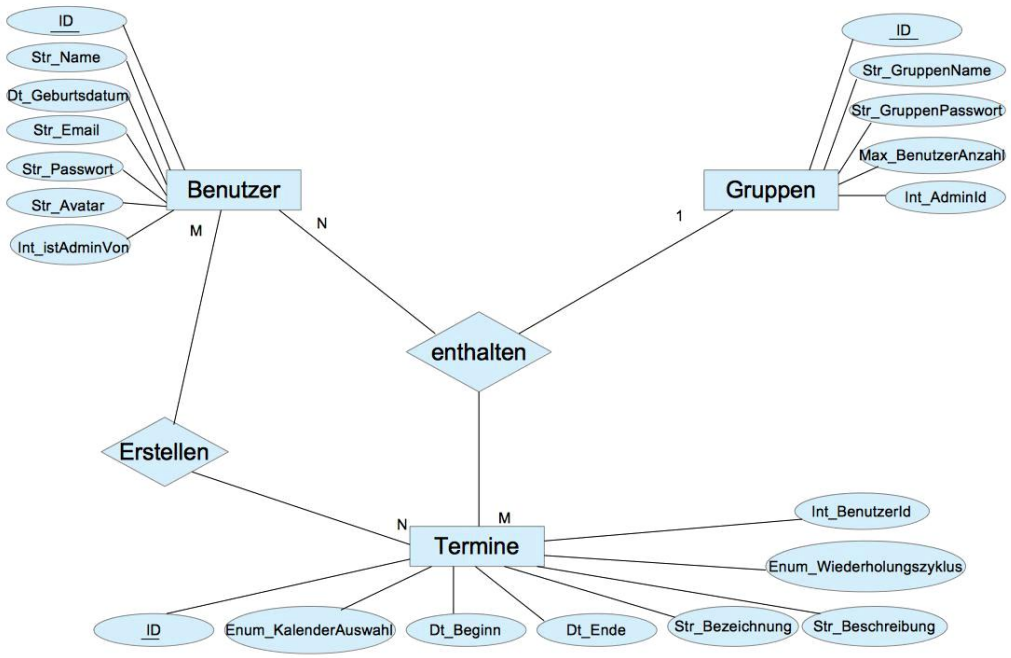
7 Prozesse und Threads (Process View)

Die Reihenfolge der Abarbeitung des Programmes erfolgt durch Vordergrund- und Hintergrundthreads. Die Hintergrundthreads werden automatisch beendet, wenn davor alle Vordergrundthreads abgeschlossen wurden. Somit wurde der Prozess zu Ende geführt.

Vordergrund-Thread:
Dieser Thread wird zum größten Teil verwendet, um den Benutzern die Benutzeroberfläche zu demonstrieren und dient der Interaktion zwischen dem Nutzer und unserem Programm. Wird vom Benutzer eine weitere Aktion zusätzlich ausgeführt, dann startet der Vordergrundthread einen neuen Hintergrundthread. Der Vordergrundthread läuft jedoch weiterhin.

Hintergrund-Thread:
Dieser Thread wird von Vordergrundthreads erzeugt und führt Programmaktionen aus. Sind diese Aktionen fertig, wird der Hintergrundthread beendet und gibt gegebenen falls die neuen Informationen an den Vordergrundthread weiter.

7 Datenspeicherung (Data View)



8 Größen und Leistung (Size and Performance)

Die Verwaltung unterstützt pro Person maximal 20.000 Einträge, da sonst durch die Einträge im Kalender zu viel Speicher in Anspruch genommen wird. Damit aufgrund des vollen Speichers die Leistung nicht zu sehr in Anspruch genommen wird, werden nach zwei Jahren alle Einträge gelöscht, die älter sind als zwei Jahre. Auf diesem Weg bleibt genügend Speicher für neue Einträge frei und die Leistung wird nicht zu stark beeinträchtigt.