

# Pflichtenheft Go-App

Grischa Hauser, Kansei Hara, Rumen Donchev, Tanja Müller, Paula Klein, Iris Landerer

22.05.2016

## Contents

<b>1</b>	<b>Zielbestimmung</b>	<b>3</b>
1.1	Musskriterien . . . . .	3
1.2	Wunschkriterien . . . . .	3
1.3	Abgrenzungskriterien . . . . .	3
<b>2</b>	<b>Produkteinsatz</b>	<b>4</b>
2.1	Anwendungsberreiche . . . . .	4
2.2	Zielgruppen . . . . .	4
2.3	Betriebsbedingungen . . . . .	4
<b>3</b>	<b>Produktumgebung</b>	<b>4</b>
3.1	Software . . . . .	4
3.2	Hardware . . . . .	4
3.3	Produktschnittstellen . . . . .	5
<b>4</b>	<b>Funktionale Anforderungen</b>	<b>5</b>
4.1	Gruppenverwaltung . . . . .	5
4.1.1	Priorität A . . . . .	5
4.1.2	Priorität B . . . . .	5
4.1.3	Priorität C . . . . .	5
4.2	Terminverwaltung . . . . .	6
4.2.1	Priorität A . . . . .	6
4.2.2	Priorität B . . . . .	6
4.2.3	Priorität C . . . . .	7
4.3	Lokalisierung . . . . .	7
4.3.1	Priorität A . . . . .	7
4.3.2	Priorität B . . . . .	8
4.3.3	Priorität C . . . . .	8
4.4	Schnittstelle zu Benutzerverwaltung . . . . .	8
4.4.1	Priorität A . . . . .	8

4.4.2	Priorität B . . . . .	8
4.4.3	Priorität C . . . . .	8
4.5	Sonstiges . . . . .	8
4.5.1	Priorität A . . . . .	8
4.5.2	Priorität B . . . . .	8
4.5.3	Priorität C . . . . .	8
4.6	Verworfenne funktionale Anforderungen . . . . .	9
4.7	Topologie . . . . .	10
4.7.1	Abhängigkeiten zwischen funktionalen Anforderungen der Gruppenverwaltung . . . . .	10
4.7.2	Abhängigkeiten zwischen funktionalen Anforderungen der Terminverwaltung . . . . .	11
4.7.3	Abhängigkeiten zwischen funktionalen Anforderungen der Lokalisierung . . . . .	12
4.7.4	Abhängigkeiten zwischen funktionalen Anforderungen der Benutzerverwaltung . . . . .	13
5	Produktdaten . . . . .	13
5.1	Personendaten . . . . .	13
5.2	Gruppendaten . . . . .	14
5.3	Terminaten . . . . .	14
6	Nichtfunktionale Anforderungen . . . . .	14
7	Globale Testfälle . . . . .	14
7.1	Folgende Funktionssequenzen sind zu überprüfen: . . . . .	14
8	Systemmodelle . . . . .	14
8.1	Szenarien . . . . .	14
8.1.1	Szenario 1 . . . . .	14
8.1.2	Szenario 2 . . . . .	15
8.1.3	Szenario 3 . . . . .	17
8.2	Anwendungsfälle . . . . .	17
8.2.1	Anwendungsfall: Kneipentour . . . . .	17
8.2.2	Anwendungsfalldiagramm . . . . .	19
8.3	Objektmodelle . . . . .	20
8.3.1	Objektmodell . . . . .	20
8.3.2	Datenbankmodell . . . . .	21
8.3.3	Servermodell . . . . .	22
8.3.4	Klassen, die das HTTP-Interface implementieren . . . . .	23
8.4	Dynamische Modelle . . . . .	24
8.4.1	Erstellen von Gruppen . . . . .	24
8.4.2	Gruppenverwaltung des Benutzers . . . . .	25
8.4.3	Erstellen eines Termins . . . . .	26
8.4.4	An- und Abmeldung von Gruppenmitgliedern zu Terminen . . . . .	28

8.4.5	Ablauf eines Termins (Veranstaltung) aus Sicht eines Teilnehmers . . . . .	29
9	Glossar	30

## 1 Zielbestimmung

Benutzer dieser App sollen dazu in die Lage versetzt werden, sich in Gruppen zu organisieren, um Treffen auszumachen und sich bei den jeweiligen Treffen unterstützt durch GPS zu finden.

### 1.1 Musskriterien

- Verwaltung der Gruppen
- Verwaltung der Termine
- Ortung der beteiligten Personen
- Schnittstelle zu Benutzerverwaltung

### 1.2 Wunschkriterien

- erweiterte Einstellungen beim Erstellen eines Termins
- alternative Benutzerverwaltung, anstatt Google-Account
- Benachrichtigungen des Benutzers über Gruppeneinladungen und wichtige Termininformationen (z.B. neu erstellt, geändert, gelöscht etc.)
- Einführung von Administratoren von einer Gruppe
- Ändern von Terminen durch den Ersteller möglich
- größere Freiheiten beim An- und Anmelden von Terminen
- Angaben, wer sich bereits am Treffpunkt befindet
- Warnungen des Benutzer bei Überschneidungen von Terminen

### 1.3 Abgrenzungskriterien

- Termine ausmachen und Teilnehmer am Treffpunkt ausfindig machen stehen im Vordergrund -> keine Funktion als Terminkalender
- keine App mit Chatfunktion
- Entwicklung vorerst nur für Android

## **2 Produkteinsatz**

Die GoApp dient zum Finden untereinander innerhalb einer Gruppe, die sich zu einem bestimmten Termin an einem Ort verabredet hat. Damit unterstützt sie die Organisation in Gruppen, das Erstellen von Terminen und die Filterung der Standorte der Gruppenmitglieder, so dass unter Beachtung der Privatsphäre einzelne Gruppenmitglieder den Rest der Gruppe finden können.

### **2.1 Anwendungsberreiche**

- Treffen mit Freunden, Kollegen etc.

### **2.2 Zielgruppen**

- jüngere, sozialaktive Menschen mit einem Android Smartphone

### **2.3 Betriebsbedingungen**

- auf einem mobilen Gerät
- unterwegs

## **3 Produktumgebung**

- Client-Server Architektur
- App läuft Clientseitig parallel mit anderen Apps auf einem mobilen Androidgerät

### **3.1 Software**

- Clientseite
  - Betriebssystem: Android 4.2 oder höher
- Serverseite
  - Tomcat 8 mit HTTP
  - MySQL-Datenbank

### **3.2 Hardware**

- Clientseite
  - Standard Android Smartphone
- Serverseite
  - Leistungsstarker Standardrechner

### 3.3 Produktschnittstellen

- Google API
- Google Maps API
- MySQL-Interface
- Schnittstelle zu GPS-Position des Android-Betriebssystems

## 4 Funktionale Anforderungen

### 4.1 Gruppenverwaltung

#### 4.1.1 Priorität A

- /F10/ Erstellen und Löschen von Gruppen
- /F20/ Gruppe einen Namen beim Erstellen geben
- /F30/ Hinzufügen und Löschen von Benutzern
- /F40/ Suchfunktion beim Erstellen, um andere Benutzer zu finden
- /F50/ Abmeldung der Gruppenmitglieder aus Gruppen
- /F60/ Übersicht über eigene Gruppen mit jeweiligen Gruppeninformationen (Gruppenname, Gruppenmitglieder)

#### 4.1.2 Priorität B

- /F70/ Benachrichtigung über Einladung zu Gruppe
- /F80/ Ersteller der Gruppe als Administrator, der Gruppe verwaltet:
  - Erstellen und Löschen von Gruppen
  - Hinzufügen und Löschen von Benutzern
  - Administrator kann sich nur aus Gruppe abmelden, wenn es weitere Administratoren gibt
- /F90/ Anzeigen des Administrators in den Gruppeninformationen

#### 4.1.3 Priorität C

- /F100/ mehrere Gruppenmitglieder können Administratorrechte auf unterschiedlichen Zugriffsebenen haben
- /F110/ Annehmen oder Absagen von Gruppeneinladungen

## 4.2 Terminverwaltung

### 4.2.1 Priorität A

- /F120/ Erstellen eines Termins mittels Angabe von
  - Terminname
  - Gruppe
  - Ort (Adresse in Textfeld eingeben)
  - Zeit
  - Datum
  - Wahl zwischen Veranstaltung und Tour (Standardeinstellung Veranstaltung)
  - Zeitraum, in dem Teilnehmer auf der Karte lokalisiert werden
    - \* Standardwert Veranstaltung: eine Stunde nach Beginn des Termins
    - \* Standardwert Tour: zwei Stunden nach Beginn des Termins
  - Notiz (optional)
- /F130/ Löschen von Terminen durch Ersteller
- /F140/ dynamische Liste für Einladungen/Anmeldungen zu Terminen für jedes Gruppenmitglied
- /F150/ Zugriff auf Termine (Teilnehmerliste, Ort, Zeit, Datum, Dauer, Notiz) nur für Gruppenmitglieder
- /F160/ An- und Abmeldung der Gruppenmitglieder zu Terminen
- /F170/ Zugriff auf Karte nur nach Anmeldung für Termine

### 4.2.2 Priorität B

- /F180/ Benachrichtigung, wenn das erste Gruppenmitglied den Treffpunkt erreicht
- /F190/ Warnung bei Anmeldung bei zwei Terminen, die gleichzeitig stattfinden
- /F200/ Hinzufügen einer Notiz bei Terminen
- /F210/ Anmeldung zu bereits abgemeldeten Terminen möglich
- /F220/ Abmeldung zu bereits angemeldeten Terminen möglich
- /F230/ Ändern von Terminen durch den Ersteller
- /F235/ Benachrichtigung, wenn Termin geändert wurde

- /F240/ Benachrichtigung, wenn neuer Termin in einer Gruppe erstellt wurde
- /F250/ Benachrichtigung über anstehende Termine am Tag des Termins
- /F260/ Liste für abgemeldete Termine für jedes Gruppenmitglied
- /F270/ Benachrichtigung über gelöschte Termine
- /F280/ Automatisches Löschen von abgelaufenen Terminen
- /F290/ Ort beim Erstellen eines Termins auf Karte suchen

#### 4.2.3 Priorität C

- /F300/ mehrere verschiedene Gruppen können sich für einen Termin zusammen verabreden

### 4.3 Lokalisierung

#### 4.3.1 Priorität A

- /F310/ Lokalisierung der Teilnehmer per GPS
- /F320/ andere Teilnehmer durch GPS-Lokalisierung auf Karte finden
- /F330/ andere Benutzer auf der Karte nur anzeigen, wenn sie in derselben Gruppe sind und dem Termin zugestimmt haben
- /F340/ GPS-Daten nur in bestimmtem Zeitraum sichtbar: ab 30 Minuten vor Beginn des Termins bis 60 Minuten (Standardeinstellung Veranstaltung) bzw. 120 min (Standardeinstellung Tour) nach Beginn des Termins
- /F350/ Karte anzeigen mithilfe von Google Maps
- /F360/ GPS-Lokalisierung ausschalten
- bei Veranstaltungen:
  - /F370/ Treffpunkt als fester Ort dargestellt
  - /F380/ Teilnehmer individuell als Marker auf der Karte dargestellt
  - /F390/ GPS-Daten nur in bestimmten Umkreis des Treffpunkts sichtbar
- bei Touren:
  - /F400/ kein fester Ort, sondern variabel, abhängig davon, wo sich Gruppe befindet
  - /F410/ größte Ansammlung an Teilnehmern an demselben Ort werden anonymisiert als Zentrum zu einem Punkt auf der Karte zusammengefasst mit dem ersten vor Ort beginnend
  - /F420/ GPS-Daten nur in bestimmten Umkreis um das Zentrum sichtbar

#### **4.3.2 Priorität B**

- /F430/ Warnung des Teilnehmers, wenn GPS aus ist
- /F440/ Angeben einer Liste von Teilnehmer, die sich bereits am Treffpunkt/im Zentrum befinden

#### **4.3.3 Priorität C**

- /F450/ Navigation zum Treffpunkt

### **4.4 Schnittstelle zu Benutzerverwaltung**

#### **4.4.1 Priorität A**

- /F460/ Registrierung des Benutzers durch Google-Account
- /F470/ Anmeldung des Benutzers
- /F480/ Speicherung der Benutzer auf dem Server
- /F490/ globales Benutzerverzeichnis auf dem Server

#### **4.4.2 Priorität B**

- /F500/ Abmeldung des Benutzers aus der App

#### **4.4.3 Priorität C**

- /F510/ Implementierung eigener Benutzerverwaltung ohne Google-Account
- /F520/ Abmeldung des Benutzers aus dem kompletten System

### **4.5 Sonstiges**

#### **4.5.1 Priorität A**

- /F530/ Datenschutzbestimmungen für Benutzer einsehbar
- /F540/ Kommunikation App <-> Server

#### **4.5.2 Priorität B**

- /F550/ Benachrichtigen deaktivieren und aktivieren durch Benutzer

#### **4.5.3 Priorität C**

- /F560/ Kommunikation zwischen Client und Server durch SSL-Verschlüsselung

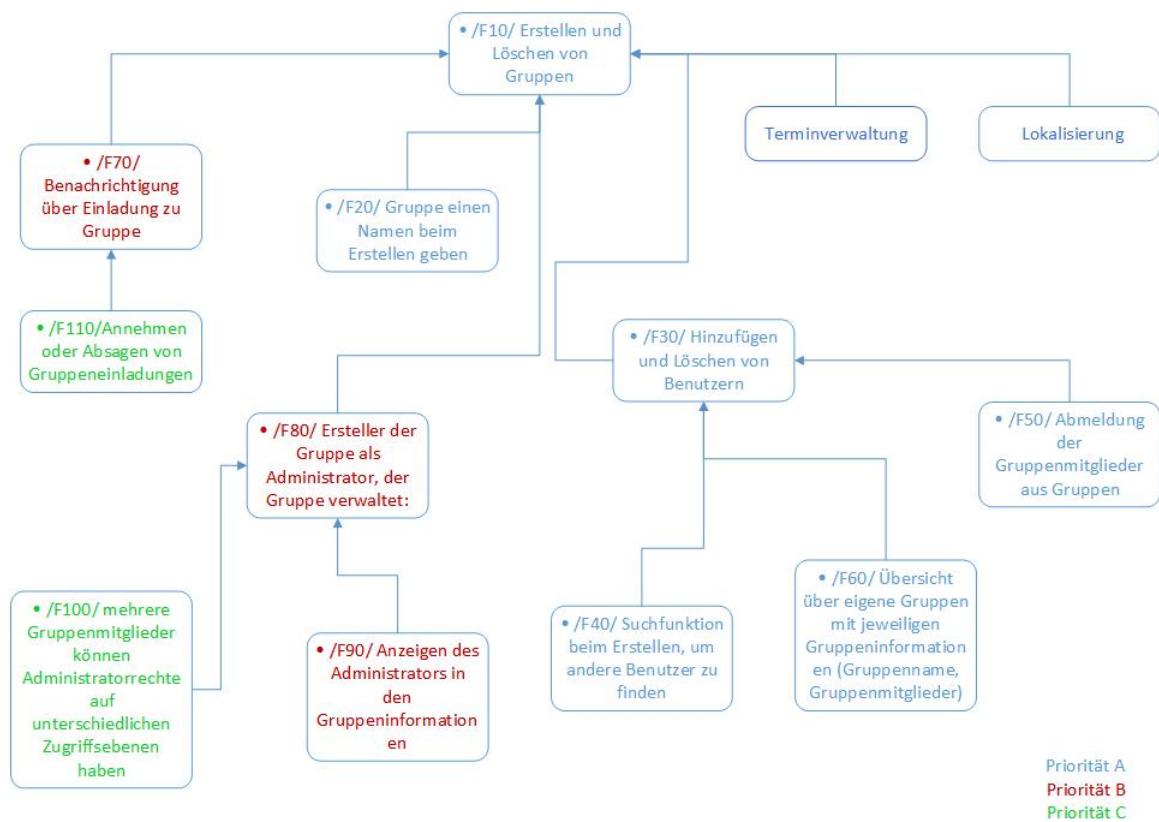


## 4.6 Verworfenne funktionale Anforderungen

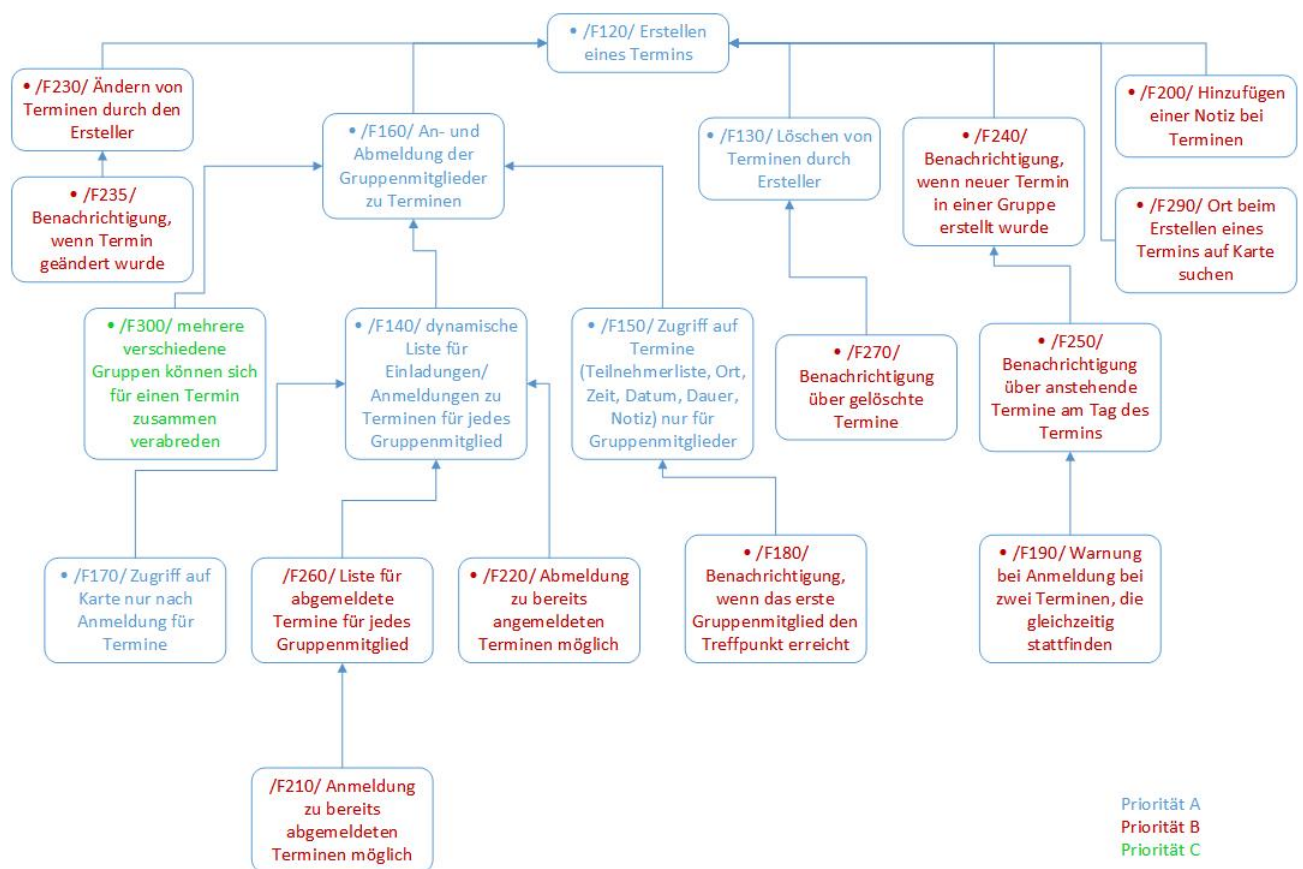
- Chat: Erfolgreiche Apps konzentrieren sich nur auf wenige Funktionen und nicht auf alle möglichen Funktionen gleichzeitig. Außerdem gibt es schon genügend Messenger-Apps.
- Anzeigen, wenn jemand der Teilnehmer im Zeitraum eines Termins keine Internetverbindung hat: Verstoß gegen Datenschutzrechte.
- Freundeskreis, um einfacher neue Gruppen zu erstellen: App ist nicht für Kommunikation und Freundefinden zuständig.
- Gruppe bei einem Termin aufteilen: nicht nötig, da man einfach einen weiteren Termin in der Gruppe erstellen kann.
- Mitteln der Lokalisierung: gemittelter Wert zu ungenau -> man möchte wissen, wo sich Gruppe genau befindet.

## 4.7 Topologie

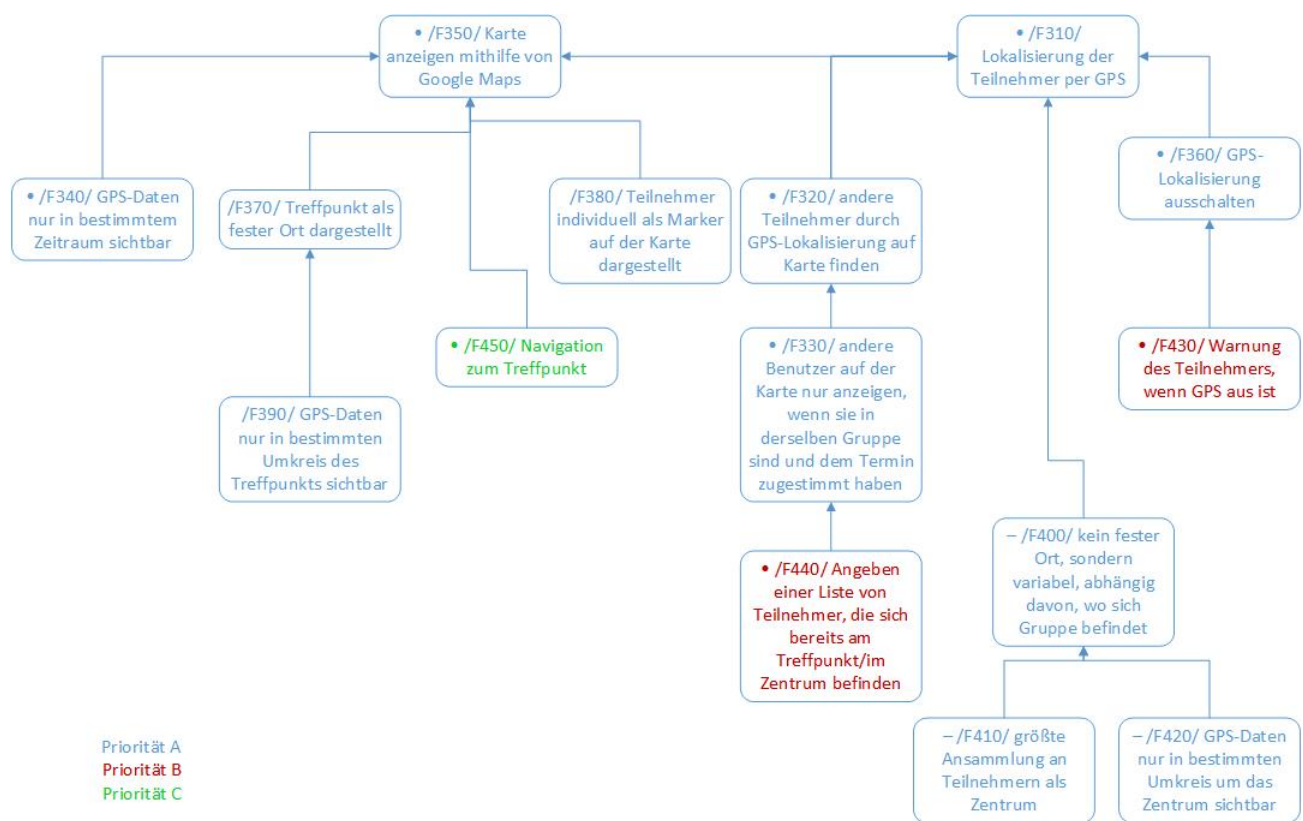
### 4.7.1 Abhängigkeiten zwischen funktionalen Anforderungen der Gruppenverwaltung



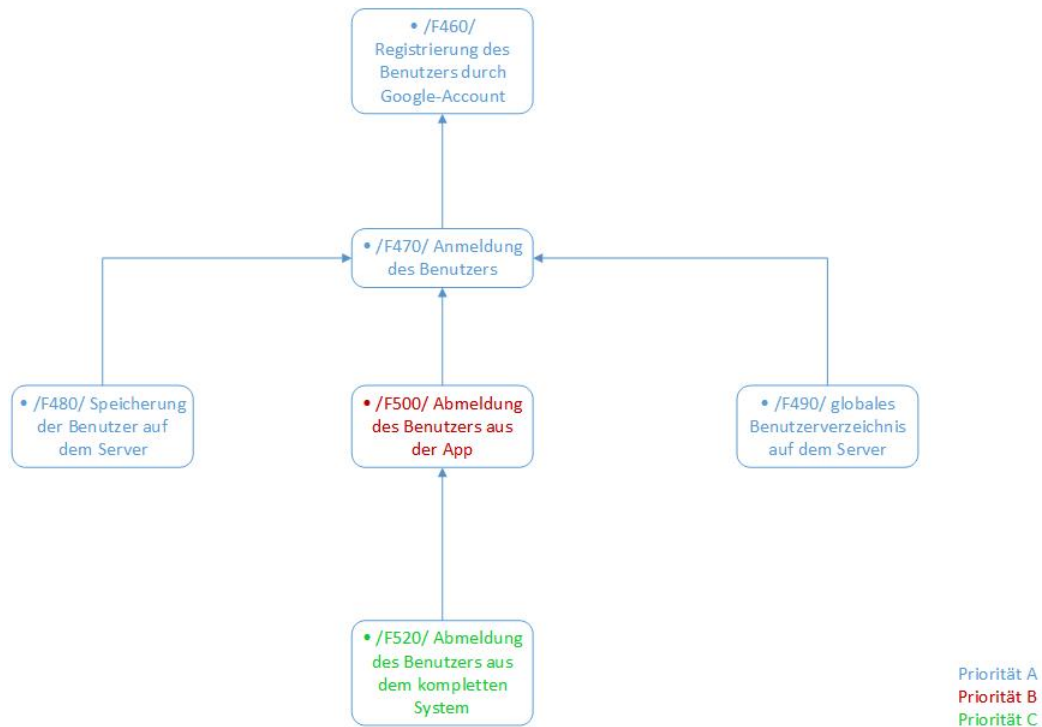
#### 4.7.2 Abhängigkeiten zwischen funktionalen Anforderungen der Terminverwaltung



#### 4.7.3 Abhängigkeiten zwischen funktionalen Anforderungen der Lokalisierung



#### 4.7.4 Abhängigkeiten zwischen funktionalen Anforderungen der Benutzerverwaltung



## 5 Produktdaten

### 5.1 Personendaten

- /D10/ Über eine Person sind folgende Daten für die Authentifizierung zu speichern: Name (Username von Google), PersonID (von Google)
- /D20/ Wenn eine Person Mitglied einer Gruppe ist, ist die GruppenID zu speichern.
- /D30/ Wenn eine Person einverstanden ist, seinen Standort zu teilen, sind seine letzten GPS-Koordinaten zu speichern.
- /D40/ Wenn man zu einem Termin eingeladen ist, ist dieser Termin zu speichern.
- /D50/ Wenn man an einem Termin teilnimmt, ist dieser Termin zu speichern.
- /D60/ Wenn man einen Termin absagt, ist dieser Termin zu speichern.

## 5.2 Gruppendaten

- /D70/ Für jede Gruppe ist die Liste aller Teilnehmer zu speichern.

## 5.3 Termindaten

- /D80/ Für jeden Termin ist eine Liste mit allen Teilnehmern zu speichern.

# 6 Nichtfunktionale Anforderungen

- /NF10/ Die Reaktionszeiten des Systems müssen unter 1 Sekunde liegen
- /NF20/ Maximal 5000 Benutzer können das System gleichzeitig benutzen
- /NF30/ Zuverlässigkeit: Nicht mehr als sechs Ausfälle pro Jahr
- /NF40/ Das System sollte in 99% der Fälle verfügbar sein

# 7 Globale Testfälle

## 7.1 Folgende Funktionssequenzen sind zu überprüfen:

# 8 Systemmodelle

## 8.1 Szenarien

### 8.1.1 Szenario 1

Eine Gruppe Studenten möchte am Abend eine Kneipentour durchführen. Sie möchte dabei nacheinander mehrere Kneipen in der Stadt besuchen. Leider können Tobias und Lisa erst später dazu kommen. Da die beiden nicht wissen, in welcher Kneipe sich die Gruppe inzwischen befindet, wenn sie dazu stoßen, beschließt die Gruppe, die Go-App zu benutzen. Sie soll Tobias und Lisa helfen, ihre Gruppe schneller zu finden.

Da die Gruppe die Go-App schon öfter eingesetzt hat, haben die Studenten die Go-App bereits auf ihrem Android-Smartphone installiert. Die Studenten sind alle Mitglieder der Gruppe "Info-Studenten".

Karl erstellt einen Termin für die Kneipentour in der Gruppe "Info-Studenten". Es öffnet sich ein Fenster, in dem Karl verschiedene Angaben zu seiner Veranstaltung machen muss. Zuerst gibt er als Terminnamen "Kneipentour" an. Nun legt er Datum, Uhrzeit und die voraussichtliche Dauer der Kneipentour fest. Gibt Alex keine Veranstaltungsdauer an, so wird die Dauer auf den Standardwert zwei Stunden gesetzt. Danach wird die Lokalisierung der Teilnehmer beendet. Falls Lisa und Tobias aber mehr als zwei Stunden später kommen, können sie nicht mehr sehen, wo sich die Gruppe aktuell befindet. Daher gibt Karl manuell eine andere Veranstaltungsdauer an. Er wählt den Button "Tour" aus, da der Veranstaltungsort der Kneipentour sich immer wieder ändert. Als

Veranstaltungsort gibt Karl in einem Textfeld die Adresse der ersten Kneipe ein, in der sich die Gruppe am Freitag Abend um 20.00 Uhr trifft. Alle weiteren Kneipen, die die Gruppe an diesem Abend besuchen will, zählt er im Feld “Notiz” auf (Priorität B).

Alle Gruppenmitglieder bekommen nun eine Benachrichtigung über die Kneipentour (Priorität B). Zusätzlich wird die Kneipentour in der chronologischen Liste der eigenen bevorstehenden Termine mit Veranstaltungsnamen und Uhrzeit angezeigt.

Als Max sich anmelden will, meldet die Go-App ihm jedoch sofort, dass er zur gleichen Zeit schon an einer anderen Veranstaltung teilnimmt (Priorität B). Max entscheidet sich für die andere Veranstaltung und sagt die Kneipentour ab. Dafür drückt er auf den Button mit “X”, der sich rechts neben dem Veranstaltungsnamen befindet. Damit verschwindet die Kneipentour aus seiner Terminliste.

Alle anderen Gruppenmitglieder sagen der Veranstaltung zu, indem sie in der Terminliste auf den Haken rechts neben der Veranstaltung drücken. Dann verschwinden die beiden Symbole zum Zu- oder Absagen und es erscheinen an dieser Stelle zwei Symbole, über die man die Google-Maps-Karte mit dem Veranstaltungsort sowie eine Liste mit den Teilnehmern direkt erreichen kann.

Nun ist es Freitag. Die Go-App erinnert alle Teilnehmer vor Beginn der Kneipentour mit einer Push-Benachrichtigung an den Termin (Priorität B). Da fällt Martin plötzlich ein, dass er an diesem Abend arbeiten muss und leider doch nicht teilnehmen kann. Obwohl er sich schon angemeldet hatte, meldet er sich nun wieder von der Kneipentour ab (Priorität B). Dazu drückt er auf die Veranstaltung und lässt sich die Veranstaltungsdetails anzeigen. Dort findet er dann einen Button, um eine bereits zugesagte Veranstaltung wieder abzusagen.

Eine halbe Stunde vor Veranstaltungsbeginn wird nun die Lokalisierung der Gruppenmitglieder aktiviert. Sobald sich mehrere Teilnehmer am gleichen Ort befinden, werden sie zu einem Zentrum zusammengefügt und als ein Punkt auf der Karte dargestellt. Jeder Teilnehmer kann nun abfragen, wer sich schon im Zentrum, also am Treffpunkt, befindet. Die Standorte der anderen Teilnehmer werden nicht angezeigt.

Als Tobias und Lisa mit einer Stunde Verspätung in die Stadt kommen, sehen sie genau, dass die Gruppe sich inzwischen in der zweiten Kneipe befindet und sie kommen sofort dort hin.

Nachdem die voraussichtliche Dauer der Veranstaltung abgelaufen ist, wird die Lokalisierung der Teilnehmer beendet und der Termin gelöscht.

### 8.1.2 Szenario 2

Am Ende eines mehrtägigen Lehrgangs möchten die Teilnehmer gemeinsam essen gehen. Zur Organisation des gemeinsamen Essens benutzen sie die Go-App.

Zunächst installieren alle Teilnehmer die Go-App auf ihrem Android-Smartphone und registrieren sich mit ihrem Google-Account.

Dann gründet Herr Schneider, der Leiter des Lehrgangs die Gruppe “Lehrgang” in der Go-App. Dabei fügt er sofort über eine Suchfunktion alle Mitglieder des

Lehrgangs zu der Gruppe hinzu.

Er erstellt für die Lehrgangsteilnehmer eine Veranstaltung, die er “Essen gehen” nennt. Dafür gibt er als Datum Freitag Abend ein und wählt den Button “Veranstaltung” aus, da die Veranstaltung an einem festen, sich nicht verändernden Ort stattfindet. Danach gibt er über ein Textfeld die Adresse des Restaurants ein.

Alle Gruppenmitglieder der Gruppe “Lehrgang” werden jetzt per Push-Benachrichtigung über die Kneipentour informiert (Priorität B) und die Veranstaltung erscheint in der eigenen Terminliste. Mit einem Klick auf die Veranstaltung werden den Teilnehmern Veranstaltungsdetails angezeigt.

Nun haben die Teilnehmer des Lehrgangs die Möglichkeit, den Termin zu- oder abzusagen, indem sie in der Terminliste rechts neben der Veranstaltung ein Kreuz oder einen Haken auswählen. Herr Maier sagt zunächst ab, da er am Freitag Abend schon zurück in seine Heimat fahren will. Die Veranstaltung ist aus seiner Terminliste verschwunden.

Am nächsten Tag überlegt er sich, doch erst am Samstag wieder nach Hause zu fahren und doch an dem Essen teilzunehmen. Er ruft in der App eine Liste mit den abgesagten Terminen auf. Dort kann er sich doch wieder zum Essen anmelden (Priorität B).

Da Herr Fischer nicht am gemeinsamen Essen teilnehmen möchte, meldet er sich aus der Gruppe “Lehrgang” sofort wieder ab.

Frau Schmidt, die erst am zweiten Tag zum Lehrgang hinzustößt, erfährt durch die anderen Teilnehmer von dem geplanten Essen am letzten Abend. Sie wendet sich an den Lehrgangsleiter, der gleichzeitig Gruppenadministrator der Gruppe “Lehrgang” ist, um in die Go-App-Gruppe aufgenommen zu werden, denn nur der Administrator kann Personen zu einer Gruppe hinzufügen oder aus einer Gruppe entfernen (Priorität B).

Da inzwischen alle Personen zu- oder abgesagt haben, ruft Herr Schneider nun in dem Restaurant an, um einen Tisch zu reservieren. Doch leider ist das Restaurant am Freitag Abend schon sehr ausgebucht und er kann keinen ausreichend großen Tisch mehr bekommen. Also reserviert er einen Tisch in einem anderen Restaurant. Dann ändert er in der Go-App-Veranstaltung den Veranstaltungsort (Priorität B). Alle Teilnehmer werden durch eine Push-Benachrichtigung darüber informiert (Priorität B).

Frau Schäfer, die schnell von Push-Benachrichtigungen genervt ist, deaktiviert diese in den Einstellungen der Go-App (Priorität B).

Nun ist es so weit. Die Veranstaltung beginnt in einer halben Stunde. Die Lokalisierung der Personen wird aktiviert und alle Teilnehmer sehen auf einer Karte die Standorte der anderen Teilnehmer. So können sie nachschauen, ob die anderen noch in ihrem Hotel oder schon unterwegs zum Restaurant sind.

Herr Maier hat den kürzesten Weg bis zum Restaurant. Da er nicht lange vor den anderen da sein möchte, schaut er, bevor er los geht, in der Go-App nach, ob die anderen schon unterwegs sind.

Der Standort von Frau Schmidt, die zwar zugesagt hat, aber jetzt doch schon nach Hause fährt, wird nicht in der Karte angezeigt, weil sie sich außerhalb eines bestimmten Radius um den Veranstaltungsort befindet.



Da die Teilnehmer sich nach dem Lehrgang wahrscheinlich nicht mehr treffen werden, löscht Herr Schneider die Gruppe am nächsten Tag.

### **8.1.3 Szenario 3**

Eine Gruppe Studenten möchte am Unifest teilnehmen. In den letzten Jahren ist es immer wieder passiert, dass einzelne Personen die Gruppe verloren haben. Stefan hat vor kurzem von der Go-App gehört, die in dieser Situation der einzelnen Person helfen soll, die Gruppe wieder zu finden. Er schlägt seiner Gruppe vor, die App beim Unifest auszuprobieren. Seine Freunde sind sofort begeistert, installieren die Go-App und registrieren sich mit ihrem Google-Account.

Stefan gründet nun eine Gruppe mit dem Namen “Party-Studenten” und fügt seine Freunde hinzu.

Dann erstellt er den Termin “Unifest”, den er als Tour deklariert, da der Veranstaltungsort zwischen verschiedenen Orten auf dem Campus wechselt. Die Gruppe will sich um 14.00 Uhr am AKK treffen. Von dort aus gehen sie über den gesamten Campus.

Jetzt erhalten alle Gruppenmitglieder eine Benachrichtigung über die Veranstaltung und sagen zu.

Doch leider verunglückt Christian wenige Tage vor dem Unifest tödlich. Christians Beerdigung ist genau am gleichen Tag wie das Unifest. Also beschließen die Studenten, nicht zum Unifest zu gehen, sondern zur Beerdigung. Stefan löscht die Veranstaltung “Unifest”.

Da das Smartphone von Christians Mutter schon ziemlich alt ist, übernimmt sie Christians Android-Smartphone. Sie entdeckt auf seinem Smartphone die Go-App und will sie auch ausprobieren. Jedoch will sie sich dafür mit ihrem eigenen Google-Account registrieren. Dazu meldet sie Christian mit seinem Google-Account von der App ab und registriert sich mit ihrem eigenen (Priorität B).

## **8.2 Anwendungsfälle**

### **8.2.1 Anwendungsfall: Kneipentour**

#### **Name**

- Kneipentour

#### **Teilnehmender Akteur**

- Gruppe von Studenten

#### **Eingangsbedingung**

- Die Studenten entscheiden, eine Kneipentour durchzuführen
- Sie beschließen, bei der Planung und Durchführung der Kneipentour die Go-App zu verwenden

- Da die Gruppe die Go-App schon öfter verwendet hat, haben alle Teilnehmer die Go-App bereits auf ihrem Android-Smartphone installiert und haben schon eine Gruppe erstellt

### **Ausgangsbedingung**

- Die Studenten haben den Abend genossen
- Sie kommen betrunken nach Hause

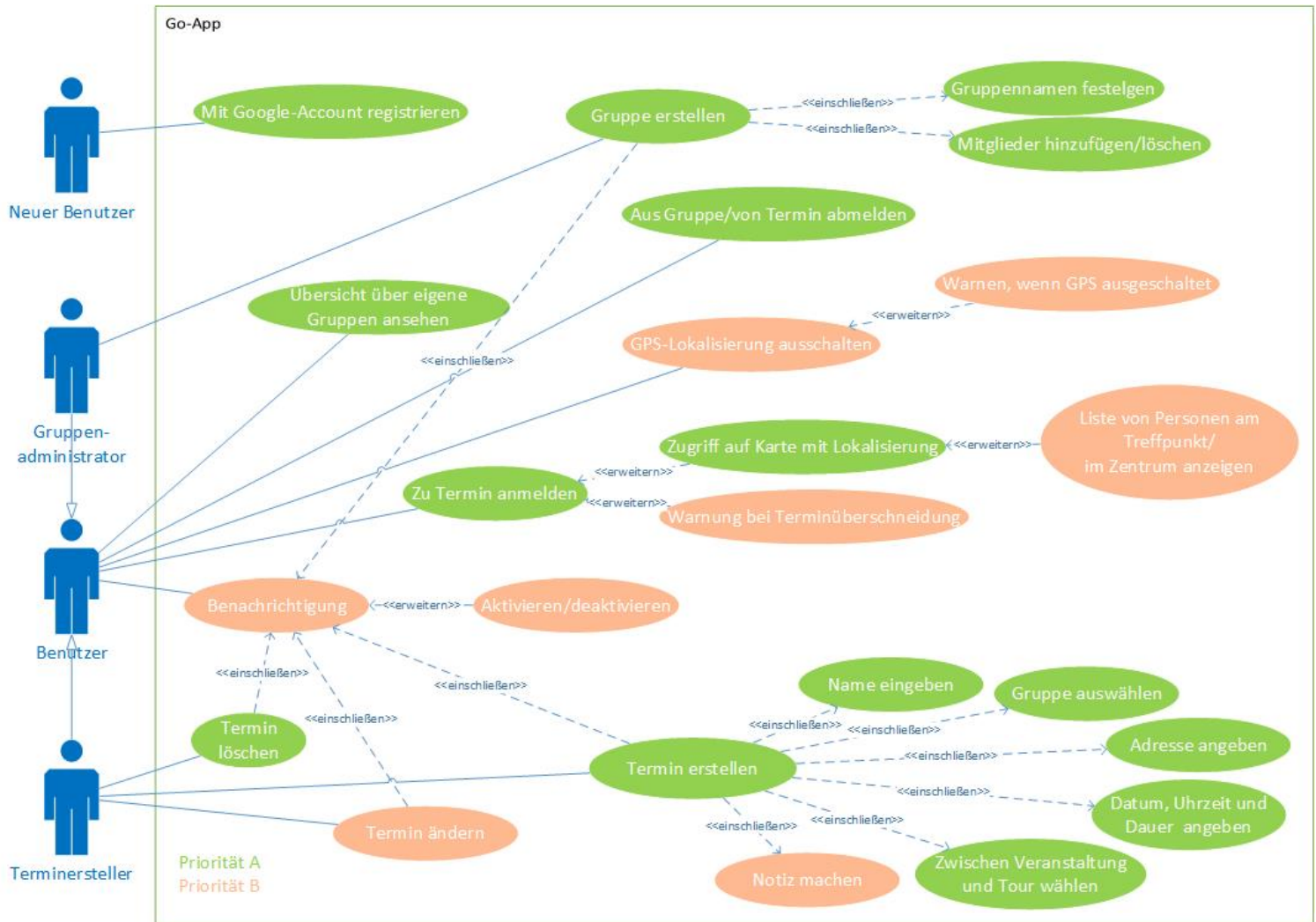
### **Ereignisfluss**

- Einer der Studenten erstellt für seine Gruppe den Termin “Kneipentour” in der Go-App unter der Angabe von Ort, Datum, Uhrzeit, Dauer, Art der Veranstaltung (Tour) und einer Liste der zu besuchenden Kneipen
- Alle Gruppenmitglieder bekommen eine Benachrichtigung über den Termin (Priorität B)
- Der Termin wird in der eigenen Terminliste angezeigt
- Die Studenten schauen sich die Veranstaltungsdetails an
- Die Studenten sagen den Termin zu oder ab
  - Bei einer Absage verschwindet die Kneipentour aus der Terminliste
  - Bei einer Zusage bekommt der Student Zugriff auf eine Teilnehmerliste
- Einzelne Studenten sagen der Kneipentour noch ab, obwohl sie anfangs zugesagt hatten und umgekehrt
- Am Veranstaltungstag erinnert die Go-App die Teilnehmer durch eine Benachrichtigung an die bevorstehende Kneipentour (Priorität B)
- Eine halbe Stunde vor Beginn der Kneipentour wird die Lokalisierung der Kneipentour aktiviert
- Sobald sich mehrere Personen in der ersten Kneipe befinden, wird ihr Standort anonymisiert auf der Karte als Zentrum dargestellt
- Personen, die zu spät kommen, können dadurch sehen, wo sich die Gruppe inzwischen befindet und direkt zur richtigen Kneipe gehen

### **Spezielle Anforderungen**

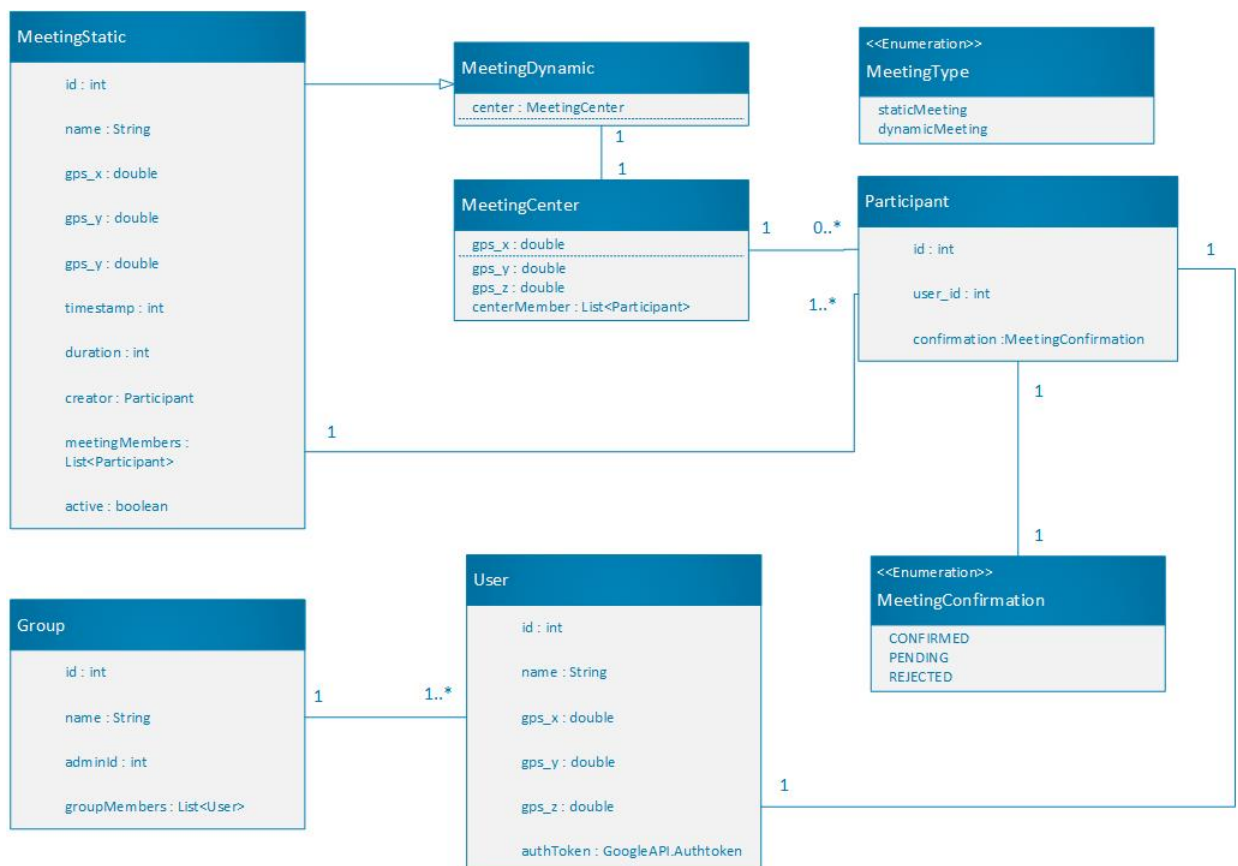
- Zugriff auf Karte mit Anzeige der Standorte bzw. des Zentrums nur nach Zusage
- Der Ort des Zentrums ist variabel, das heißt das Zentrum ist immer genau dort, wo sich die größte Personengruppe befindet

## 8.2.2 Anwendungsfalldiagramm

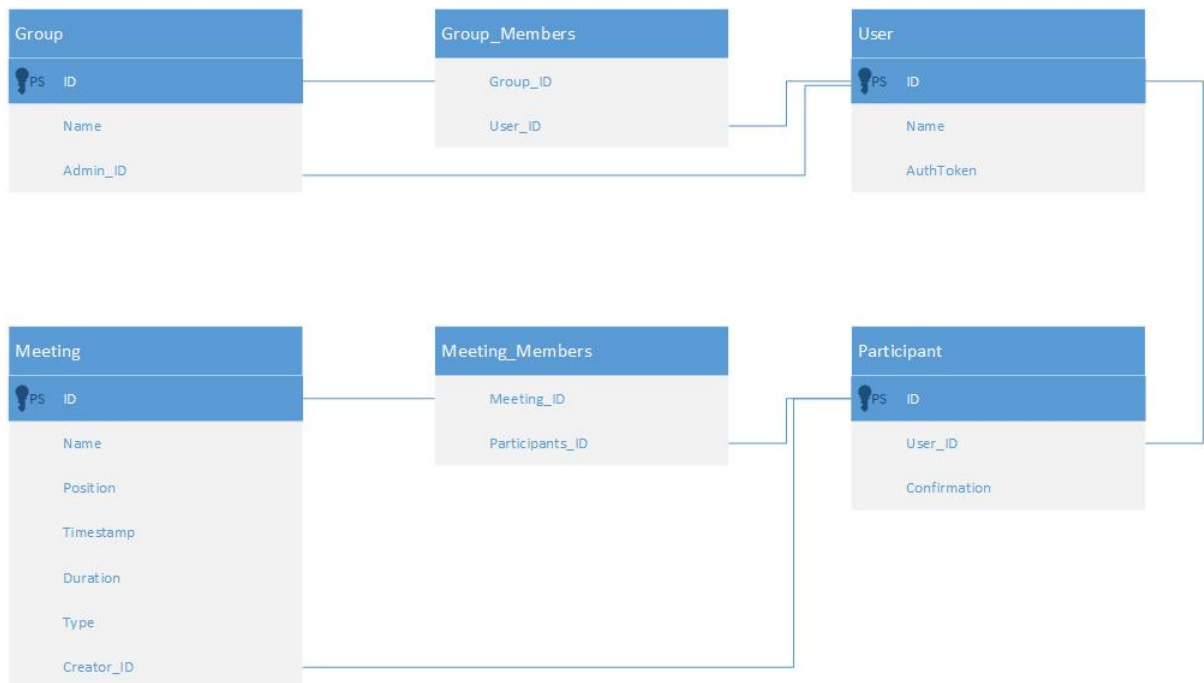


## 8.3 Objektmodelle

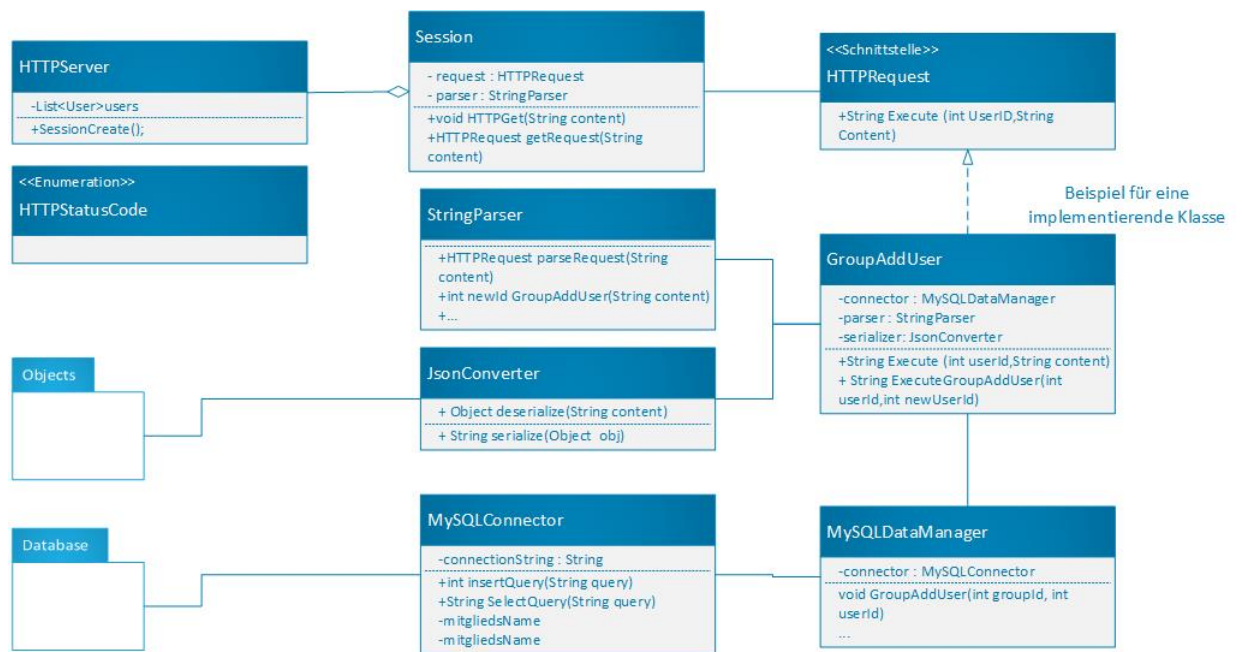
### 8.3.1 Objektmodell

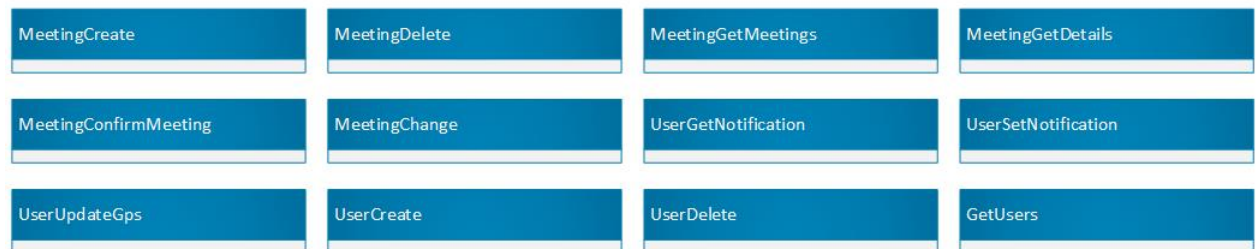


### 8.3.2 Datenbankmodell



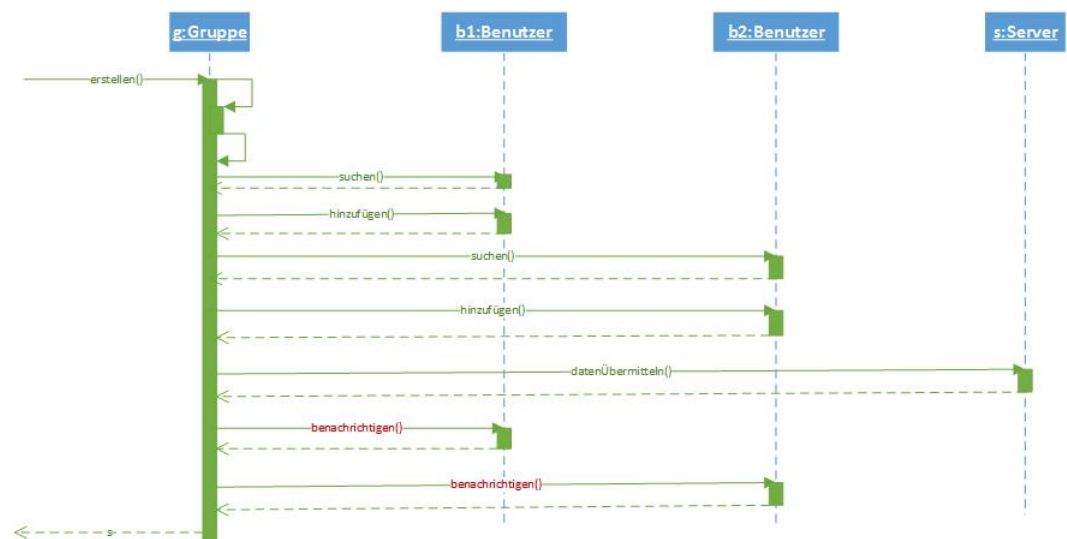
### 8.3.3 Servermodell





## 8.4 Dynamische Modelle

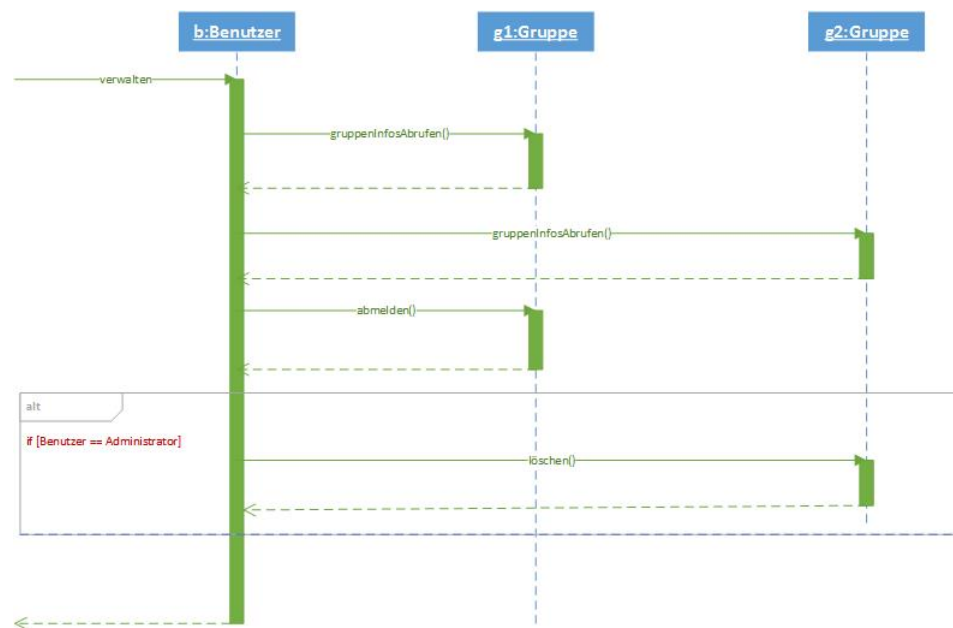
### 8.4.1 Erstellen von Gruppen



Funktionale Anforderungen mit Priorität B



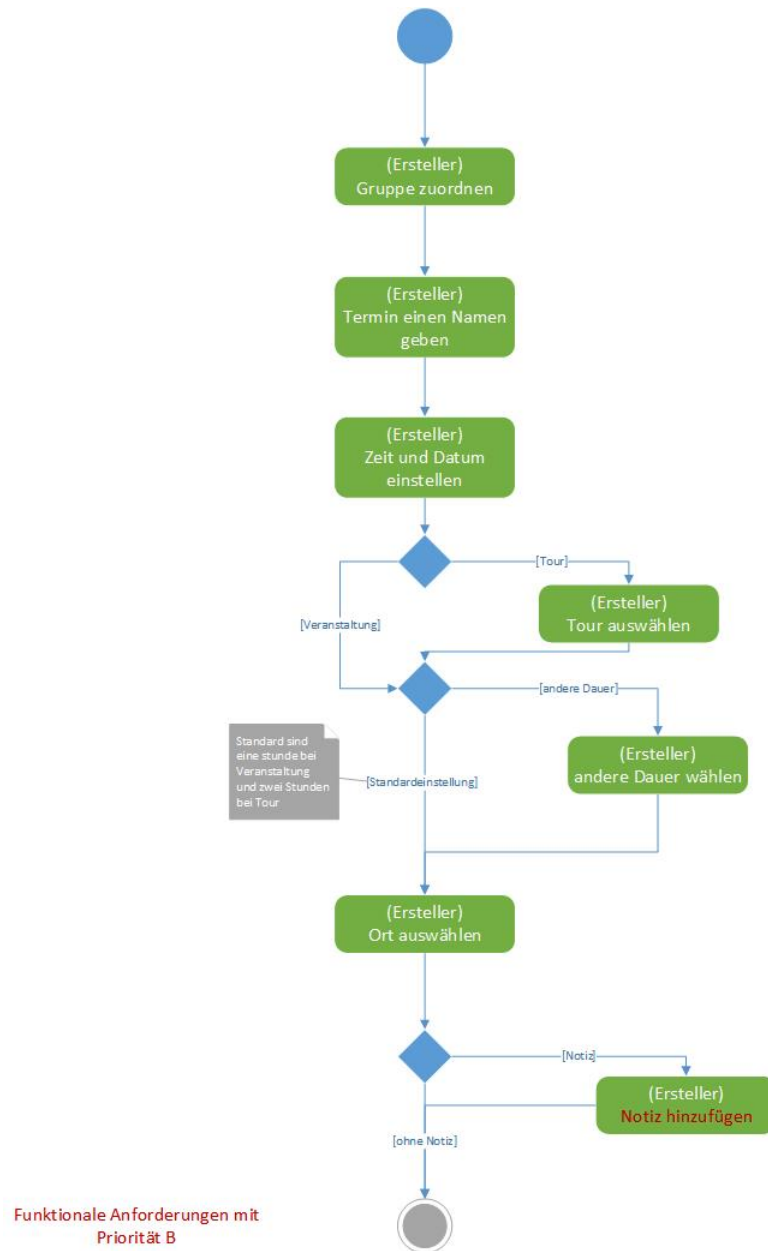
#### 8.4.2 Gruppenverwaltung des Benutzers



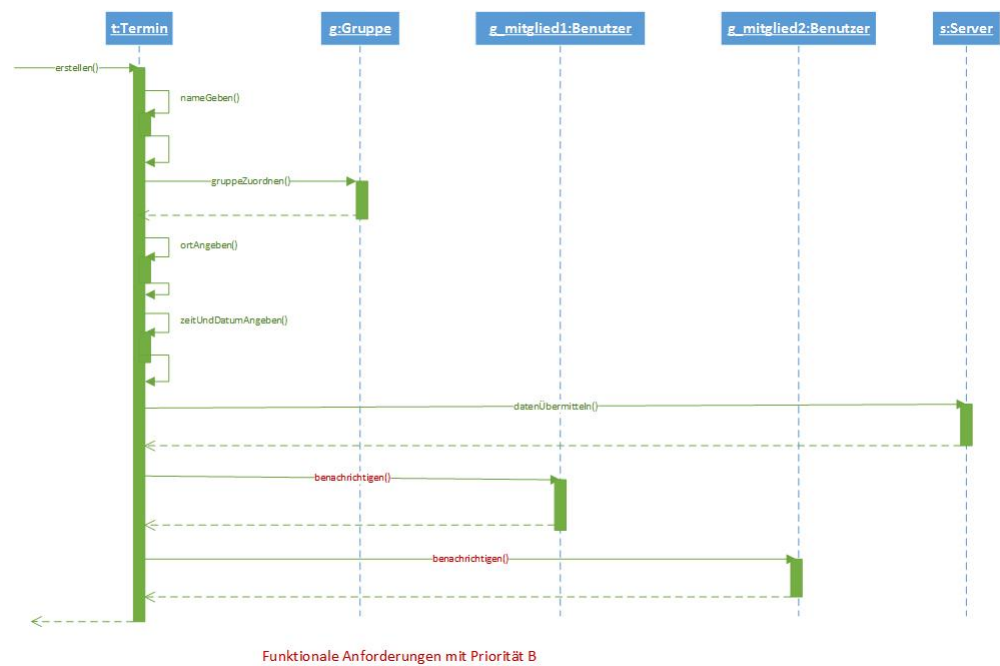
Funktionale Anforderungen mit Priorität B

### 8.4.3 Erstellen eines Termins

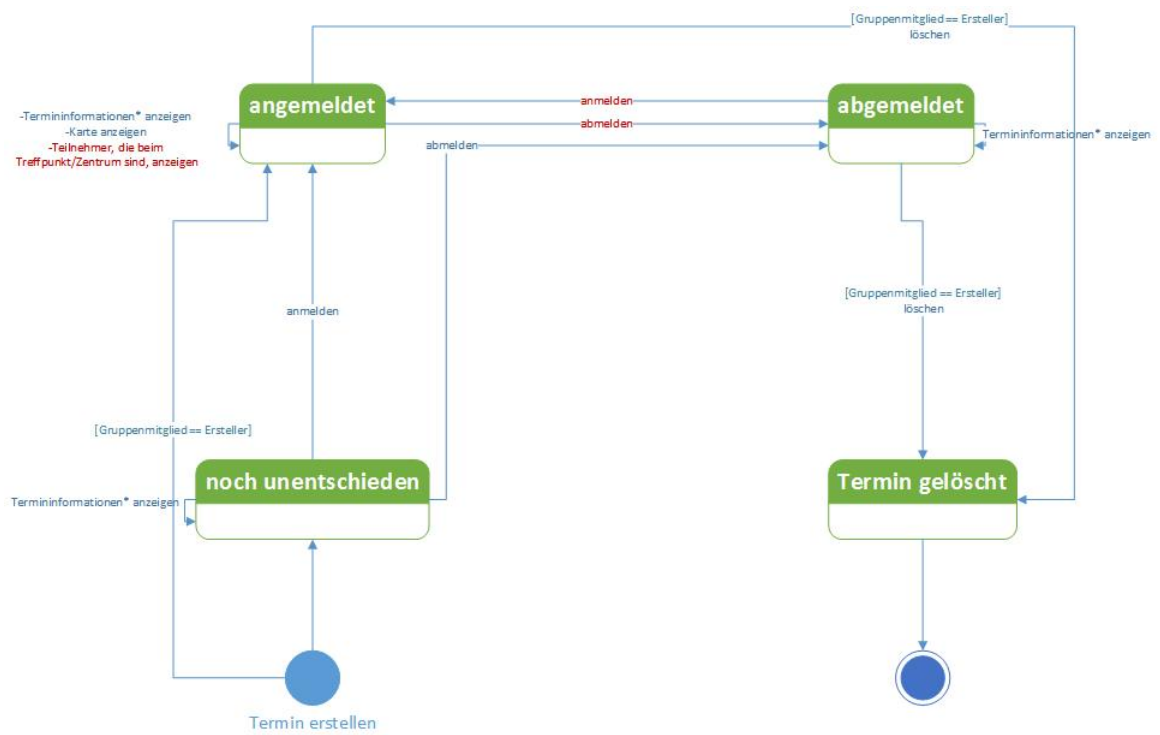
#### Aktivitätsdiagramm



## Sequenzdiagramm



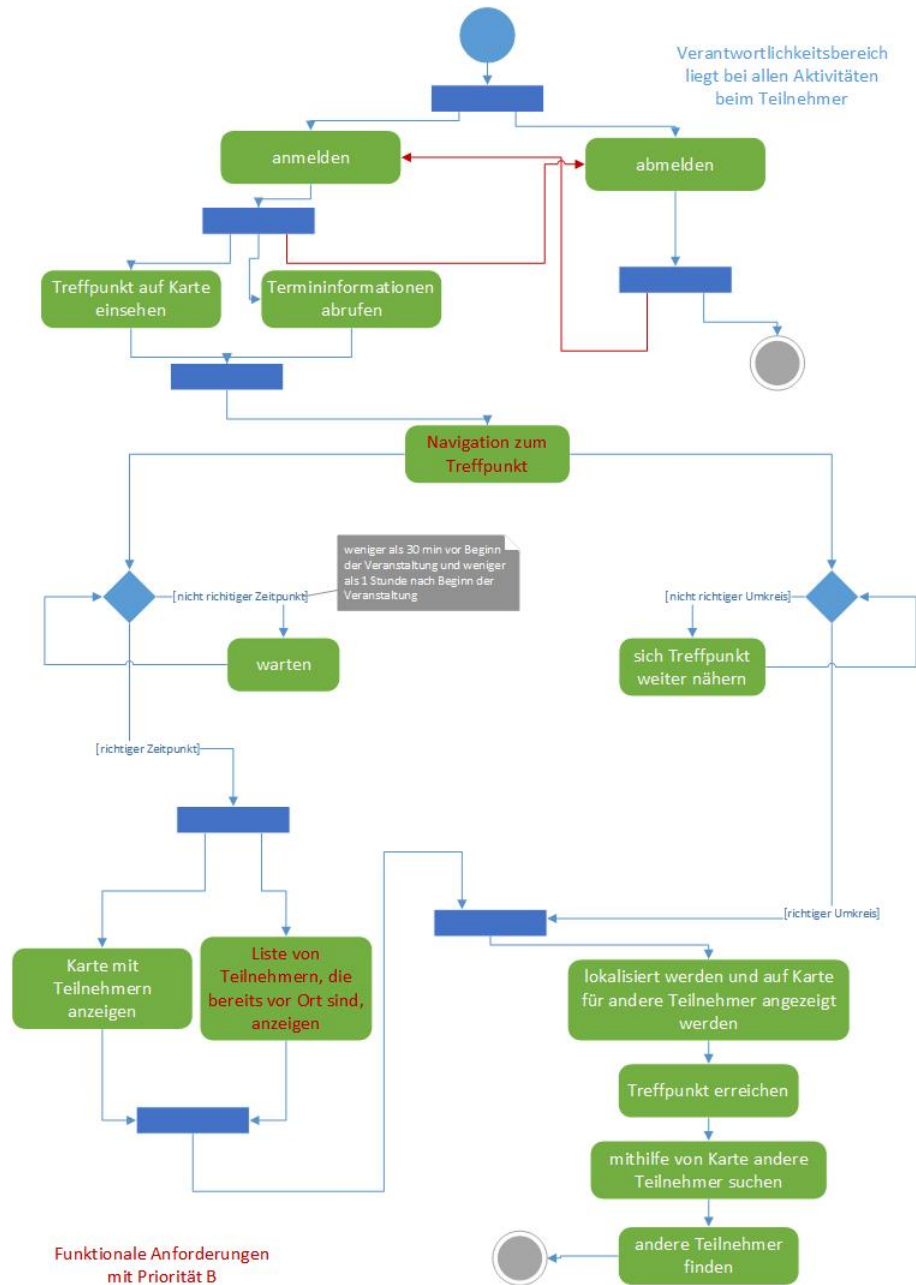
#### 8.4.4 An- und Abmeldung von Gruppenmitgliedern zu Terminen



Termininformationen\* : Teilnehmer, Ort, Zeit, Datum, Dauer, Notiz

Funktionale Anforderungen mit Priorität B

#### 8.4.5 Ablauf eines Termins (Veranstaltung) aus Sicht eines Teilnehmers



## 9 Glossar