

# Testbericht

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

21. August 2016

## Inhaltsverzeichnis

<b>1</b>	<b>Korrigierte Fehler und Bugs</b>	<b>2</b>
1.1	Server . . . . .	2
1.2	Client . . . . .	3
<b>2</b>	<b>Veränderungen</b>	<b>4</b>
<b>3</b>	<b>Testabdeckung</b>	<b>4</b>
3.1	Client . . . . .	4
3.2	Server . . . . .	5
<b>4</b>	<b>Szenarien/Manuelle Tests</b>	<b>5</b>
4.1	Gruppenverwaltung . . . . .	5
4.2	Terminverwaltung . . . . .	5
4.3	Lokalisierung . . . . .	6
4.4	Schnittstelle zur Benutzerverwaltung . . . . .	6

# 1 Korrigierte Fehler und Bugs

## 1.1 Server

Symptom	Grund	Behebung
Exception, wenn eine Liste aller Gruppen eines Users angefordert wird, der Benutzer aber in keiner Gruppe ist	Ungefangene Exception in der Methode <code>getAllGroups():List&lt;Group&gt;</code> in der Klasse <code>GroupDaoImpl</code>	Wenn ein User in keiner Gruppe Mitglied ist, wird eine leere Liste zurückgegeben
Ein User kann zu einer Gruppe hinzugefügt werden, obwohl dieser User nicht existiert	Es wird nicht geprüft, ob der User in der Datenbank ist, bevor er zu einer Gruppe hinzugefügt wird	Vor dem Hinzufügen eines Users zu einer Gruppe prüfen, ob der User existiert
Ein User kann zu einem Meeting hinzugefügt werden, obwohl dieser User nicht existiert.	Es wird nicht geprüft, ob der User in der Datenbank ist, bevor er zu einem Meeting hinzugefügt wird	Vor dem Hinzufügen eines Users zu einem Meeting prüfen, ob der User existiert
Ein User, der bereits in einer Gruppe ist, kann nochmal in dieselbe Gruppe hinzugefügt werden	Es wird nicht geprüft, ob der User schon in der Gruppe ist, bevor er zu der Gruppe hinzugefügt wird	Vor dem Hinzufügen eines Users zu einer Gruppe wird geprüft, ob der User schon in der Gruppe ist
User, der bereits Participant eines Meetings ist, kann nochmal zu dem Meeting hinzugefügt werden	Es wird nicht geprüft, ob der User schon Participant des Meetings ist	Vor dem Hinzufügen eines Users zu einem Meeting wird geprüft, ob der User schon Participant des Meetings ist
User können in einer Gruppe upgedatet oder aus einer Gruppe gelöscht werden, in der sie kein Gruppenmitglied sind	Es wird nicht geprüft, ob der User überhaupt Mitglied der Gruppe ist	Vor dem Updaten oder Löschen eines Users aus einer Gruppe wird geprüft, ob der User Mitglied der Gruppe ist
User können in einem Meeting upgedatet oder aus dem Meeting gelöscht werden, obwohl sie kein Participant des Meetings sind	Es wird nicht geprüft, ob der User überhaupt Participant des Meetings ist	Vor dem Updaten oder Löschen eines Users aus einem Meeting wird geprüft, ob der User Participant des Meetings ist
Confirmation des Creators eines Meetings ist auf "pending" gesetzt, sollte aber "confirmed" sein, da der Creator an seinem eigenen Termin teilnehmen sollte	Confirmation eines Participants ist bei default immer auf "pending" gesetzt	Confirmation des Creators ist bei default immer „confirmed“. Alle anderen Participants haben bei default "pending" als Confirmation
Beim Updaten eines Meetings kommt die Fehlermeldung, dass ein Meeting einen Creator braucht	Es gibt eine Überprüfung, ob die <code>creatorId &gt;=0</code> ist. Da die <code>creatorId</code> nicht im <code>jsonString</code> übergeben wird, ist die <code>creatorId</code> nicht <code>&gt;=0</code> .	Die Überprüfung, ob die <code>creatorId</code> vorhanden ist, wurde in der Methode <code>updateMeeting(DatabaseConection)</code> der Klasse <code>MeetingDaoImpl</code> entfernt, da ein Meeting immer einen Creator hat und dieser nicht upgedatet werden kann
Der Server hat sehr häufig die Session des Clients verloren	Nicht genau bekannt. Zusammenspiel zwischen Tomcat und Android <code>httpappclient</code>	Neues session system basierend auf Cookies geschrieben. Mit eigener klasse zum managen von Cookies inklusive Anpassungen im login system

## 1.2 Client

Symptom	Grund	Behebung
Error: Klassenname AboutActivity existierte doppelt	Klasse AboutActivity wurde in das Package Service kopiert, obwohl sie dort nicht hin gehört (Fehler beim Pushen in Github)	Klasse AboutActivity aus dem Package Service gelöscht
Error: Klassenname GroupMemberActivity existierte doppelt	Klasse GroupMemberActivity wurde in das Package Service kopiert, obwohl sie dort nicht hin gehört (Fehler beim Pushen in Github)	Klasse GroupMemberActivity aus dem Package Service gelöscht
Error: Klassenname GroupsActivity existierte doppelt	Klasse GroupsActivity wurde in das Package Service kopiert, obwohl sie dort nicht hin gehört (Fehler beim Pushen in Github)	Klasse GroupsActivity aus dem Package Service gelöscht
Creator eines Meetings kann sich selber als Participant zu seinem Meeting hinzufügen (wurde beim Server und beim Client gefixt)	Fehlende Filterung des eigenen Users	UserID wird beim Login aus „SharedPreferences“ gespeichert und beim Erstellen eines Meetings gefiltert
Creator einer Gruppe kann sich selber als Gruppenmitglied zu seiner Gruppe hinzufügen	Fehlende Filterung des eigenen Users	UserID wird beim Login aus „SharedPreferences“ gespeichert und beim Erstellen einer Gruppe gefiltert
Beim Erstellen eines Meetings werden sofort alle Participants an den Server übermittelt. Der Server speichert jedoch nur den Creator und verwirft alle anderen Participants	Server speichert die Participants erst, wenn sie nach dem Erstellen des Meetings separat übermittelt werden	Das Meeting wird beim Client erstellt und dem Server übermittelt. Danach werden bei einer separaten Anfrage die Participants übermittelt.
Beim Hinzufügen von Participants erhält man badRequest (Errorcode 400)	Im Intent war der JsonString im falschen Schlüssel gespeichert	Der Schlüssel „CommunicationKey.Meeting“ wird zu „CommunicationKey.Participant“ geändert
Nachdem ein Termin abgesagt wurde, verschwindet ProgressDialog nicht mehr	Es wurde Statuscode 202 erwartet, statt 200	Statuscode 202 wurde zu 200 geändert und es wurde ein default status hinzugefügt, welches den ProgressDialog schließt
Wenn GPS ausgeschaltet ist, gibt es eine NullPointerException	Exception nicht behandelt	In der Methode getGPS wurde eine if-Abfrage hinzugefügt, ob GPS eingeschaltet ist
Es werden alle Participants eines Meetings angezeigt deren Status „confirmed“ oder „pending“ ist. Es sollen jedoch nur die Participants angezeigt werden, deren Status „confirmed“ ist	In der MeetingParticipantActivity war die Bedingung zum Hinzufügen eines Participants zu einem Meeting „!= rejected“	Die Bedingung wurde zu „== confirmed“ geändert

Bei GPS-Koordinaten (double-Format) stürzt die CreateNewMeetingActivity ab	Die CreateNewMeetingActivity versucht, die GPS-Daten von String in long zu parsen	String in double umgewandelt
Exception beim Ausführen der Klasse android.os.NetworkOnMainThreadException beim HTTP.execute	BroadcastReceiver verhindert, dass Aktionen im Internet ausgeführt werden	AlarmReceiver startet den GPS-Service
AlarmReceiver konnte nicht gestartet werden	AlarmReceiver war im Manifest falsch definiert (Schreibfehler)	Schreibfehler wurde korrigiert
RuntimePermission bei Android 6	Methode getLocation() in der Klasse AlarmReceiver für Android 6 nicht angepasst	Methode getLocation() neu geschrieben
GPS-Koordinaten : Breitengrad kann auch außerhalb des Intervalls [-90°,90°) und Längengrad außerhalb der Werte [-180°,180°) eingegeben werden	Fall wurde in der Implementierung nicht behandelt	Fehlermeldung, wenn die Werte außerhalb des Intervalls liegen
Länge des Namens für User, Group und Meeting nicht begrenzt	Länge wurde in der Implementierung nicht festgelegt	Maximale Länge des Namens für User, Group und Meeting wurde auf 20 Zeichen festgelegt
Auf der Karte wurde der Treffpunkt falsch angezeigt (MapActivity)	Latitude und Longitude wurden beim Übergeben verwechselt	Parameter für Latitude und Longitude wurden richtig eingesetzt

## 2 Veränderungen

- Database erweitert: DatabaseAdapter für die Activitys geschrieben, der die Methoden insertMeeting(), getMeeting() und deleteRow() der Klasse DataBaseHandler vereinfacht. Die innere Klasse DataBaseHandler implementiert die Methoden onCreate(SQLiteDatabase) und onUpgrade(SQLiteDatabase, int, int) der Klasse SQLiteDatabase
- MeetingListActivity erweitert: MeetingListActivity verwendet Methoden der Klasse DatabaseAdapter (z.B. insertMeeting())
- CreateNewMeetingActivity: Suchfeld für Suchadresse und Button zum automatischen Hinzufügen von Koordinaten hinzugefügt
- Wenn ein User in keiner Gruppe Mitglied ist, kann er beim Erstellen eines Meetings keine Gruppe auswählen aber bekam auch keine Warnung (benutzerunfreundlich). Jetzt bekommt er eine Warnmeldung und kommt automatisch zurück zur MeetingListActivity
- Wenn ein User keine bevorstehenden Meetings hat, wurde ihm eine leere Liste angezeigt. Es wurde ein Feature hinzugefügt, das in diesem Fall statt der Tabelle Meldung „Keine bevorstehenden Termine“ anzeigt

## 3 Testabdeckung

### 3.1 Client

Da wir uns in dieser Phase hauptsächlich darauf konzentriert haben, die Bugs und Fehler aus Implementierungsphase zu fixen, hatten wir am Ende nicht mehr so viel Zeit, um JUnit-Tests zu schreiben. Außerdem sind beim Schreiben der JUnit-Tests in einigen Methoden Exceptions aufgetreten (NullPointerException), obwohl diese bei manuellen Tests funktionieren. Daher vermuten wir, dass diese

Methoden nur manuell getestet werden können und haben daraufhin unsere (nicht funktionierenden) JUnit-Tests wieder aus dem Projekt entfernt. Deshalb wurde im gesamten Projekt auf der Clientseite nur eine Testabdeckung von 34% der Klassen erreicht.

Im Package `edu.kit.pse.client.goapp.converter` wurde die größte Überdeckung mit 100% der Klassen und 92% der Zeilen erreicht.

Das Package `edu.kit.pse.client.goapp.datamodels` hat eine Abdeckung der Klassen von 100% und der Zeilen von 60%.

Bei den Uri-Buildern wurden die Klassen zu 92% überdeckt und die Zeilen zu 90%.

Im Package `edu.kit.pse.client.goapp.activity` gab es einige Schwierigkeiten beim mit JUnit-Tests. Diese Klassen wurden hauptsächlich manuell getestet. Daher wurde nur eine Testüberdeckung von 34% der Klassen erreicht.

Die anderen Packages konnten teilweise nicht direkt getestet werden, da sie einige private oder protected Methoden enthalten. Die verbleibenden Klassen und Methoden lassen sich, wie oben schon erwähnt, vermutlich nur manuell testen.

## 3.2 Server

Beim Server konnten wir hingegen eine deutlich höhere Testabdeckung erzielen. Die gesamte Testabdeckung des Servers beträgt 78,7%.

Die Packages `kit.edu.pse.goapp.server.algorithm` und `kit.edu.pse.goapp.server.exceptions` erreichten sogar eine Überdeckung von 100%.

Die Packages `kit.edu.pse.goapp.server.converter.dao` wurde zu 97,6% überdeckt und das Package `kit.edu.pse.goapp.server.converter.object` zu 91%.

Das Package `kit.edu.pse.goapp.server.creating_obj_with_dao` wurde zu 90,5% überdeckt.

Im Package `kit.edu.pse.goapp.server.tests` konnte eine Testabdeckung von 89,8% erreicht werden.

Das Package `kit.edu.pse.goapp.server.validation` erreichte eine Überdeckung von 87,1%.

Im Package `kit.edu.pse.goapp.server.datamodels` wurde eine Überdeckung von 84,8% erreicht.

Das Package `kit.edu.pse.goapp.server.daos` konnte zu 80,9% überdeckt werden.

Das Package `kit.edu.pse.goapp.gson.typeadapters` wurde zu 77% mit Tests überdeckt.

Das Package `kit.edu.pse.goapp.server.authentication` erreichte eine Testabdeckung von 72,7%.

Und das Package `kit.edu.pse.goapp.server.servlets` konnte zu 31,7% abgedeckt werden.

## 4 Szenarien/Manuelle Tests

Folgende Funktionen der App wurden manuell getestet:

### 4.1 Gruppenverwaltung

- Erstellen einer Gruppe mit Gruppennamen
- Hinzufügen von Personen zu der Gruppe
- Anzeigen der Gruppeninformationen: Namen der Gruppenmitglieder
- Löschen eines Gruppenmitgliedes
- Gruppe löschen

### 4.2 Terminverwaltung

- Erstellen eines Termins mit Angabe von: Terminname, Gruppe, Ort, Zeit, Datum und Dauer der Veranstaltung
- Anzeigen von Daten des Termins: Datum, Uhrzeit, Terminname, Ort und Ersteller des Termins
- Zusage oder Absagen eines Termins

- Zugriff auf Teilnehmerliste und Karte, nachdem man einen Termin zugesagt hat
- Automatisches Löschen eines Termins, wenn er vorbei ist

### **4.3 Lokalisierung**

- Lokalisierung aller Teilnehmer einer Veranstaltung eine halbe Stunde vor Beginn des Termins
- Warnung an Teilnehmer mit ausgeschaltetem GPS
- Anzeige der Teilnehmer als individuelle Marker auf der Karte ist nur für Gruppenmitglieder sichtbar, sie am Termin teilnehmen, sobald sie sich in einem bestimmten Umkreis um den Treffpunkt befinden
- Die GPS-Lokalisierung wird nach Ablauf der Veranstaltungsdauer beendet

### **4.4 Schnittstelle zur Benutzerverwaltung**

- Registrierung eines Benutzers über den Google-Account
- Anmelden des Benutzers in der App
- Speicherung des Benutzers auf dem Server