

Android-Go-App Testbericht

Dennis Bäuml,
Theresa Heine,
Victoria Karl,
Tarek Wilkening

Betreuer:

Erik Burger,
Axel Busch,
Heiko Klare

12. März 2017

Inhaltsverzeichnis

1	Einleitung	3
2	Testvorbereitung	4
2.1	Zu testende Komponenten	4
2.2	Verwendete Frameworks	4
2.2.0.1	Mockito und PowerMockito	4
2.2.0.2	Espresso	4
2.2.0.3	Google Android Lint	5
3	Automatische Tests	6
3.1	Unit Tests	6
3.1.0.1	Client Controller ObjectStructure	6
3.1.0.2	Client Model	7
3.1.0.3	Server Controller	8
3.1.0.4	Server Model	8
3.1.0.5	Server Communication	9
3.2	Android Tests	11
3.2.0.1	Client Controller Database	11
3.3	Integration Tests	13
4	Manuelle Tests	16
4.1	Testszzenarien	16
4.1.0.1	Kneipentour	16
4.1.0.2	Mensa	16
4.2	UI Tests	16
5	Testabdeckung	17
5.1	Clientseitig	17
5.2	Serverseitig	17
6	Verbleibende Fehler	18
6.1	Fehler Behebungen	18
6.2	Verbliebene Fehler	18

1 Einleitung

Die Android-Go-App erleichtert es Nutzern sich in Gruppen zu organisieren und Treffpunkte zu vereinbaren. Sobald der Zeitpunkt des Treffens näher rückt, können alle Mitglieder ihre GPS-Daten an die anderen Gruppenmitglieder übermitteln und diese darüber informieren, wie weit sie sich vom Zielort entfernt befinden.

Dieses Dokument beschreibt dabei die Durchführung der Testphase der einzelnen Module und Applikationen, um eine möglichst störungsfreie Benutzung der Android-Go-App zu gewährleisten. Dabei werden auf die im Pflichtenheft definierten Testfälle und Szenarien eingegangen.

Zusätzlich wurden die in der Implementierungsphase in Verzug geratenen Unit Tests nachgeholt, als auch die Hauptfunktion (das Versenden, Empfangen, Aktualisieren und Anzeigen der GPS-Standorte in der Kartenansicht) der Android-Go-App implementiert. Funktionen wie Benutzer und Gruppen löschen als auch weitere Wunschkriterien wurden, wie nach der Implementierungsphase besprochen, nicht nachträglich hinzugefügt, um eine erfolgreiche Testphase zu ermöglichen.

2 Testvorbereitung

2.1 Zu testende Komponenten

Clientseitig wurden Model und Controller ausführlich getestet. Bei der View wurde die GUI auf Stabilität überprüft und versucht den Teil, welcher dem Controller entspricht, so gut wie möglich zu testen. Dabei wurde in erster Linie Wert darauf gelegt, dass die Receiver, welche die Antworten des Servers abfangen und auswerten problemlos funktionieren. Zusätzlich wurde der Code optimiert, damit die Go-App schneller und stabiler läuft.

Serverseitig wurden die Requests und Responses getestet, welche ja äquivalent zu denen vom Client sind also nicht zweimal getestet werden mussten. Zusätzlich wurde der GroupServer ausführlich getestet.

Funktionen welche zu großen Teilen aus generiertem Code bestehen wurden nicht einzeln getestet, sondern nur in ihrer Interaktion mit anderen Anwendungen. Das Model der Android Datenbank, welches nur aus Klassen mit Attributen und keinerlei Funktion besteht, wurde nicht getestet.

2.2 Verwendete Frameworks

2.2.0.1 Mockito und PowerMockito

Das Mockito bzw. PowerMockito Framework haben wir für Unit Tests verwendet, um kleine Einheiten des Codes unabhängig von anderen Teilen des Programms zu testen. So konnte sichergestellt werden, dass die Einheiten an sich funktionieren und auftretende Fehler beseitigen.

2.2.0.2 Espresso

Espresso ist ein UI Test Framework mit welchem UI Tests an einem Emulator oder Gerät ausgeführt werden können. Es erleichtert zusätzlich das Testen von Android spezifischen Klassen, welche mit Mockito gar nicht oder nur schwer zu testen sind.

2.2.0.3 Google Android Lint

Google Android Lint ist ein Optimierungs- Framework für Android Applikationen um sicherzustellen, dass der Code keine strukturellen Probleme aufweist. Es überprüft die Android-Projekt-Quelldateien auf mögliche Bugs und Optimierungsverbesserungen für Korrektheit, Sicherheit, Leistung, Benutzerfreundlichkeit und Zugänglichkeit. Durch Verwendung dieses Frameworks wird unser Code optimiert, indem Terme vereinfacht und redundanter Code entfernt werden. Damit ist die App besser wartbar, läuft schneller und stürzt nicht mehr so leicht ab. Wird also im Allgemeinen stabiler in ihrer Anwendung.

3 Automatische Tests

3.1 Unit Tests

3.1.0.1 Client Controller ObjectStructure

Testmethode	Testbeschreibung	Ergebnis
TestRegisterUser	Testet ob ein RegistrationRequest mit den richtigen Informationen über den Benutzer an den Server gesendet wird.	✓
TestDeleteUser	Testet ob ein DeleteUserRequest mit der Device Id an den Server gesendet wird.	✓
TestCreateGroup	Testet ob ein CreateGroupRequest mit dem neuen Gruppennamen an den Server geschickt wird.	✓
TestJoinGroup	Testet ob ein JoinGroupRequest mit dem Link und der Device Id an den Server geschickt wird.	✓
TestDeleteGroup	Testet ob ein DeleteGroupRequest mit dem Gruppennamen an den Server geschickt wird.	✓
TestGetDeviceID	Testet ob eine gültige Device id vom Gerät abgerufen wird.	✓

Tabelle 1: AccountHandlerTest und GroupHandlerTest

3.1.0.2 Client Model

Testmethode	Testbeschreibung	Ergebnis
TestOnCreate	Testet ob die Datenbank mit ihren zwei Tabellen erstellt wird.	✓
TestOnUpgrade	Testet ob die alte Datenbank gelöscht und dann eine neue aufgebaut wird.	✓
TestToSimpleAppointment	Testet ob ein Appointment erfolgreich zu einem SimpleAppointment gecastet wurde.	✓
TestSetAppointmentDate	Testet ob Datum und Uhrzeit für ein Treffen gesetzt wurden.	✓
TestSetDestination	Testet ob Zielort und GPS-Koordination für ein Treffen gesetzt wurden.	✓
TestCreateInviteLink	Testet ob ein CreateLinkRequest mit der Device Id und dem Gruppennamen an den Server geschickt wird.	✓
TestGetGroupUpdate	Testet ob ein GroupUpdateRequest mit der Device Id und dem Gruppennamen an den Server geschickt wird.	✓
TestMakeGroupMemberToAdmin	Testet ob ein MakeAdminRequest mit der Device Id, der Member Id und dem Gruppennamen an den Server geschickt wird.	✓
TestGetAllGroupMemberNames	Testet ob alle Gruppenmitglieder einer Gruppe aufgezählt werden.	✓
TestDeleteGroupMember	Testet ob ein KickMemberRequest mit der Device Id, der Member Id und dem Gruppennamen an den Server geschickt wird.	✓
TestLeaveGroup	Testet ob ein LeaveGroupRequest mit der Device Id und dem Gruppennamen an den Server geschickt wird.	✓
TestGetMemberTypeTrue	Testet ob für den Gruppenadministrator aus der Android Datenbank ausgelesen wird, dass er Admin ist.	✓
TestGetMemberTypeFalse	Testet ob für ein GruppenMitglied aus der Android Datenbank ausgelesen wird, dass es nicht Administrator ist.	✓
TestActivateGoService	Testet ob die Android Datenbank aktualisiert wird, sobald der GoStatus aktiviert wird.	✓
TestDeactivateGoService	Testet ob die Android Datenbank aktualisiert wird, sobald der GoStatus deaktiviert wird.	✓
TestChangeGroupName	Testet ob ein RenameGroupRequest mit der Device Id an den Server geschickt wird.	✓
TestLinkToString	Testet ob die vom Server erhaltenen Informationen für den Link zu einen String umgewandelt werden können.	✓

Tabelle 2: DBGroupHandler-, Appointment-, GroupClient und LinkTest

3.1.0.3 Server Controller

Testmethode	Testbeschreibung	Ergebnis
TestGetUser	Testet ob man ein SimpleUser Objekt von der Datenbank bekommt.	✓
TestGetGroup	Testet ob man ein GroupServer Objekt von der Datenbank bekommt.	✓
TestPersistObject	Testet ob ein SimpleUser Objekt in der Datenbank gespeichert wird.	✓
TestPersistObject	Testet ob ein GroupServer Objekt in der Datenbank gespeichert wird.	✓
TestDeleteObject	Testet ob ein SimpleUser Objekt von der Datenbank gelöscht wurde.	✓
TestDeleteObject	Testet ob ein GroupServer Objekt von der Datenbank gelöscht wurde.	✓

Tabelle 3: ResourceManagerTest

3.1.0.4 Server Model

Testmethode	Testbeschreibung	Ergebnis
TestAddAdmin	Testet ob ein Benutzer als Gruppenadministrator einer Gruppe hinzugefügt wird.	✓
TestCreateLink	Testet ob ein Link erstellt wird und dieser auch die richtigen Informationen über die Gruppe und das Secret enthält.	✓
TestJoinGroup	Testet ob ein Benutzer einer Gruppe über einen Link betreten kann und wenn der Link nochmal verwendet wird, dass das nicht geht.	✓
TestRemoveMember	Testet ob ein Gruppenmitglied aus einer Gruppe entfernt wird.	✓
TestGetGpsData	Testet ob die GPS Daten für einen Benutzer gesetzt werden.	✓
TestGetMemberAssociations	Testet ob man alle Informationen über den Status der Gruppenmitglieder erhält.	✓

Tabelle 4: GroupServerTest

3.1.0.5 Server Communication

Testmethode	Testbeschreibung	Ergebnis
TestStartServer	Testet ob der Server gestartet wird.	✓
TestShutDown	Testet ob der Server gestoppt wird.	✓
TestSendRequest	Testet ob ein HTTP Request gesendet wurde.	✓
TestGetRequest	Testet ob man eine HTTP Request bekommt.	✓
TestRegistration	Testet ob eine RegistrationResponse auf eine RegistrationRequest erstellt wird.	✓
TestDeleteUser	Testet ob eine DeleteUserResponse auf eine DeleteUserRequest erstellt wird.	✓
TestRenameUser	Testet ob eine RenameUserResponse auf eine RenameUserRequest erstellt wird.	✓
TestCreateGroup	Testet ob eine CreateGroupResponse auf eine CreateGroupRequest erstellt wird.	✓
TestDeleteGroup	Testet ob eine DeleteGroupResponse auf eine DeleteGroupRequest erstellt wird.	✓
TestCreateLin	Testet ob eine CreateLinkResponse auf eine CreateLinkRequest erstellt wird.	✓
TestJoinGroup	Testet ob eine JoinGroupResponse auf eine JoinGroupRequest erstellt wird.	✓
TestKickMember	Testet ob eine KickMemberResponse auf eine KickMemberRequest erstellt wird.	✓
TestLeaveGroup	Testet ob eine LeaveGroupResponse auf eine LeaveGroupRequest erstellt wird.	✓
TestMakeAdmin	Testet ob eine MakeAdminResponse auf eine MakeAdminRequest erstellt wird.	✓
TestRenameGroup	Testet ob eine RenameGroupResponse auf eine RenameGroupRequest erstellt wird.	✓
TestNegativeRenameGroup	Testet ob auf eine RenameGroupRequest mit einem nicht eindeutigen Namen eine Response mit success = false erstellt wird.	✓
TestSetAppointment	Testet ob eine SetAppointmentResponse auf eine SetAppointmentRequest erstellt wird.	✓
TestUpdate	Testet ob eine UpdateResponse auf eine UpdateRequest erstellt wird.	✓

Tabelle 5: Request Test

3.2 Android Tests

3.2.0.1 Client Controller Database

Testmethode	Testbeschreibung	Ergebnis
TestInsertNewGroup	Testet ob eine neue Gruppe in die Gruppentabelle der Android Datenbank hinzugefügt wird.	✓
TestReadOneGroupRow	Testet ob über den Gruppennamen die richtige Gruppe aus der Gruppentabelle der Android Datenbank ausgelesen wird.	✓
TestReadAllGroupNames	Testet ob alle Gruppennamen aus der Gruppentabelle der Android Datenbank ausgelesen werden.	✓
TestDeleteAllGroups	Testet ob alle Gruppen aus der Gruppentabelle der Android Datenbank gelöscht werden.	✓
TestDeleteOneGroupRow	Testet ob eine Gruppe aus der Gruppentabelle der Android Datenbank gelöscht wird.	✓
TestUpdateGroupDate	Testet ob der Eintrag einer Gruppe in der Gruppentabelle der Android Datenbank angepasst wird, sobald sich das Appointment, der GoStatus oder der Gruppenname ändert.	✓
TestInserUserData	Testet ob ein neues Mitglied einer Gruppe der Benutzertabelle der Android Datenbank hinzugefügt wird.	✓
TestReadAllGroupMembers	Testet ob alle Mitgliedernamen einer Gruppe aus der Benutzertabelle der Android Datenbank ausgelesen werden.	✓
TestReadAllGroupMemberIds	Testet ob alle Mitglieder-Id's einer Gruppe aus der Benutzertabelle der Android Datenbank ausgelesen werden.	✓
TestReadAdminOrMemberStatus	Testet ob der richtige Admin oder Member Status aus der Benutzertabelle der Android Datenbank eines Mitglieds zu einer Gruppe ausgelesen wird.	✓
TestDeleteAllUserAndGroups	Testet ob alle Mitglieder und zugehörigen Gruppen aus der Benutzertabelle der Android Datenbank gelöscht werden.	✓
TestDeleteUserFromGroup	Testet ob ein Mitglied einer Gruppe aus der Benutzertabelle der Android Datenbank gelöscht wird.	✓
TestDeleteAllGroupMembers	Testet ob alle Mitglieder einer Gruppe aus der Benutzertabelle der Android Datenbank gelöscht werden.	✓
TestUpdateGroupNameInAlloc	Testet ob der Gruppenname einer Gruppe für alle Mitglieder in der Benutzertabelle der Android Datenbank angepasst wird, wenn dieser geändert wurde..	✓
TestUpdateUserName	Testet ob der Benutzername vom Aktuellen Benutzer in allen Gruppen in denen er Mitglied ist, in der Benutzertabelle der Android Datenbank angepasst wird, wenn er seinen Benutzernamen ändert.	✓

Tabelle 6: GroupServiceTest und UserServiceTest

Testmethode	Testbeschreibung	Ergebnis
TestUpdateGroup-MemberToAdmin	Testet ob ein Gruppenmitglied in der Benutzertabelle der Android Datenbank zu einem Administrator derjenigen Gruppe gemacht wird.	✓

Tabelle 7: GroupServiceTest und UserServiceTest

3.3 Integration Tests

Testfall	Testmethode	Testbeschreibung	Ergebnis
/T010/	Benutzer registrieren	Wenn der Benutzer mit dem Internet verbunden ist und der gewählte Benutzernamen gültig ist, wird er auf dem Server angelegt und danach in den Shared Preferences gespeichert.	✓
/T020/	Benutzer registrieren Fehlschlag 1	Wenn der Benutzer mit dem Internet verbunden ist aber der gewählte Benutzernamen ungültig ist, wird er nicht auf dem Server angelegt und somit auch nicht in den Shared Preferences.	✓
/T030/	Benutzer registrieren Fehlschlag 2	Wenn der Benutzer nicht mit dem Internet verbunden ist wird er nicht auf dem Server angelegt und somit auch nicht in den Shared Preferences.	✓
/T040/	Benutzeraccount löschen	Wenn der Benutzer mit dem Internet verbunden ist und auswählt seinen Account zu löschen, dann wird dieser vom Server gelöscht als auch alle seine Daten über ihn und seine Gruppen von der Android Datenbank.	✓
/T050/	Benutzeraccount löschen Fehlschlag	Wenn der Benutzer nicht mit dem Internet verbunden ist, wird er weder vom Server noch von der Android Datenbank gelöscht.	✓
/T060/	Gruppe erstellen	Wenn der Benutzer mit dem Internet verbunden und der Gruppenname gültig ist, dann wird die Gruppe mit ihm als einziges Mitglied (als Administrator) auf dem Server und in der Android Datenbank angelegt.	✓
/T070/	Gruppe erstellen Fehlschlag	Wenn der Benutzer nicht mit dem Internet verbunden ist und eine Gruppe erstellt, dann wird diese weder auf dem Server noch auf der Android Datenbank angelegt.	✓
/T080/	Gruppe löschen	Wenn der Benutzer mit dem Internet verbunden ist und auswählt eine Gruppe zu löschen, dann wird diese und alle ihre Mitglieder vom Server als auch von der Android Datenbank gelöscht.	✓
/T090/	Gruppe löschen Fehlschlag	Wenn der Benutzer nicht mit dem Internet verbunden ist, wird die Gruppe und ihre Mitglieder weder vom Server noch von der Android Datenbank gelöscht.	✓
/T100/	Gruppenlink erstellen	Wenn der Benutzer Administrator und mit dem Internet verbunden ist und ein Mitglied einladen möchte, dann erhält er einen Link, welcher auf dem Server in Zusammenhang mit der jeweiligen Gruppe gespeichert wird.	✓
/T110/	Gruppenlink erstellen Fehlschlag	Wenn der Benutzer Administrator aber nicht mit dem Internet verbunden ist, dann erhält er keinen Link und es wird auch keiner in Zusammenhang mit der Gruppe auf dem Server gespeichert.	✓

Tabelle 8: Testfälle aus dem Pflichtenheft

Testfall	Testmethode	Testbeschreibung	Ergebnis
/T120/	Gruppe über Link beitreten	Wenn der Benutzer mit dem Internet verbunden und bereits bei der Go-App registriert ist, dann wird der Link mit der App geöffnet, der Benutzer der Gruppe hinzugefügt, auf dem Server und der Android Datenbank gespeichert und der Link vom Server gelöscht.	✓
/T130/	Gruppe über Link beitreten Fehlschlag	Wenn der Benutzer bereits bei der Go-App registriert aber nicht mit dem Internet verbunden ist, dann wird der Benutzer der Gruppe nicht hinzugefügt und der Link nicht vom Server gelöscht.	✓
/T140/	Gruppe verlassen	Wenn der Benutzer mit dem Internet verbunden ist und auswählt eine Gruppe zu verlassen, dann wird er auch als Mitglied dieser Gruppe vom Server und die gesamte Gruppe von der Android Datenbank gelöscht.	✓
/T150/	Gruppe verlassen Fehlschlag	Wenn der Benutzer nicht mit dem Internet verbunden ist und auswählt eine Gruppe zu verlassen, dann wird er nicht als Mitglied dieser Gruppe vom Server und der Android Datenbank gelöscht.	✓
/T160/	Mitglied aus Gruppe entfernen	Wenn der Benutzer Administrator und mit dem Internet verbunden ist und auswählt ein Mitglied zu löschen, dann wird dieses aus der Gruppe entfernt und vom Server als auch der Android Datenbank gelöscht.	✓
/T170/	Mitglied aus Gruppe entfernen Fehlschlag	Wenn der Benutzer Administrator aber nicht mit dem Internet verbunden ist und auswählt ein Mitglied zu löschen, dann wird dieses nicht der Gruppe entfernt und auch weder vom Server noch von der Android Datenbank gelöscht.	✓
/T180/	Zielort festlegen	Wenn der Benutzer Administrator, mit dem Internet verbunden ist und einen Zielort auswählt, dann wird der Zielort der Gruppe auf dem Server und der Android Datenbank aktualisiert.	✓
/T190/	Uhrzeit festlegen	Wenn der Benutzer Administrator, mit dem Internet verbunden ist und eine Uhrzeit für den Treffpunkt wählt, dann wird die Uhrzeit der Gruppe auf dem Server und der Android Datenbank aktualisiert.	✓
/T200/	Zielort/ Uhrzeit festlegen Fehlschlag	Wenn der Benutzer Administrator, nicht mit dem Internet verbunden ist und Zielort und/oder Uhrzeit auswählt, dann wird die Gruppe weder auf dem Server noch in der Android Datenbank aktualisiert.	✓
/T210/	Go-Button drücken	Wenn der Benutzer mit dem Internet verbunden ist und den Go-Button für eine Gruppe drückt, dann wird dies in der Android Datenbank angepasst und sein Standort alle 15 Sekunden auf dem Server aktualisiert und an die anderen Gruppenmitglieder weitergeleitet.	✓

Tabelle 9: Fortsetzung Testfälle aus dem Pflichtenheft

Testfall	Testmethode	Testbeschreibung	Ergebnis
/T220/	Go-Button drücken Fehlschlag	Wenn der Benutzer nicht mit dem Internet verbunden ist und den Go-Button drückt, dann wird dieser weder in der Android Datenbank angepasst noch sein Standort an den Server übermittelt und dort gespeichert.	✓
/T230/	GPS-Daten auf Karte anzeigen	Wenn der Benutzer den Go-Button aktiviert und mit dem Internet verbunden ist, dann erhält er alle 15 Sekunden die Standorte der anderen Gruppenmitglieder. Diese werden auf der Karte zu einem Punkt zusammengefasst, wenn sie nahe beieinander liegen.	✓
/T240/	Verschlüsselung zwischen Client und Server	Wenn der Benutzer mit dem Internet verbunden ist und Daten an den Server übermittelt/ empfängt, dann sind diese verschlüsselt.	✓
/T250/	Go-Button deaktivieren	Wenn der Benutzer seinen Go-Button nochmal drückt nachdem er diesen aktiviert hat, dann wird sein Standort nicht mehr an den Server übermittelt und die Android Datenbank aktualisiert.	✓
/T260/	Gruppenparameter aktualisieren	Wenn der Benutzer auf eine Gruppe drückt, dann wird die Gruppe, also ihre Mitglieder und der Treffpunkt auf dem Server und der Android Datenbank aktualisiert.	✓
/T270/	Gruppenparameter aktualisieren Fehlschlag	Wenn der Benutzer auf eine Gruppe drückt aber nicht mit dem Internet verbunden ist, dann werden die Gruppenparameter weder in der Android Datenbank noch auf dem server aktualisiert.	✓

Tabelle 10: Zweite Fortsetzung der Testfälle aus dem Pflichtenheft

4 Manuelle Tests

4.1 Testszenarien

4.1.0.1 Kneipentour

4.1.0.2 Mensa

4.2 UI Tests

5 Testabdeckung

5.1 Clientseitig

Paketname	abgedeckte Methoden	abgedeckte Linien in %
controller.communication		%
controller.obejctStructure	7/7	95%
controller.database	18/18	87%
model.database	3/3	83%
model.objectStructure	90/100	92%
view		%
Gesamt (über alle Pakete)		%

Tabelle 11: Testüberdeckung Clientseitig

5.2 Serverseitig

Package name	Covered Methods	Covered Lines%
		%
		%
		%
		%
		%
		%
		%
		%
Gesamt (über alle Pakete)		%

Tabelle 12: Testüberdeckung Serverseitig

6 Verbleibende Fehler

6.1 Fehler Behebungen

Clientseitig

1. Die Receiver in der View identifizieren eine Response einer Request durch den Namen der jeweiligen Request. Zuvor konnte es vorkommen, dass unter Umständen ein anderer Receiver die Response abgefangen hat, die Response nicht auslesen konnte und einen Fehler geworfen hat.
- 2.
- 3.
- 4.
- 5.

Serverseitig

- 1.
- 2.
- 3.
- 4.

6.2 Verbliebene Fehler

1. Bei GroupUpdate kommt es manchmal zu Problemen, wenn die Gruppe oder Mitglieder aus dieser Gruppe gelöscht wurden.
2. Wenn man GUI Buttons mehrfach drückt, bevor die Aktion ausgeführt wurde, kann es zu Fehlern kommen.
- 3.
- 4.

5.

6.

Weitere Fehler sind nicht bekannt.