



DELAY TOLERANT JODEL

Software-Entwicklungspraktikum (SEP)
Sommersemester 2016

Abnahmetestspezifikation

Auftraggeber
Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Prof. Dr.-Ing. Lars Wolf
Mühlenpfordtstraße 23
38106 Braunschweig

Betreuer: Björn Gernert & Dominik Schürmann

Auftragnehmer:

Name	E-Mail-Adresse
Philip Wunram	p.wunram@tu-braunschweig.de
Alexey Sivokha	a.sivokha@tu-braunschweig.de
Ali Güzelkaya	a.guezelkaya@tu-braunschweig.de
Florian Rampold	f.rampold@tu-braunschweig.de
Lukas Schmidt	lukas.schmidt@tu-braunschweig.de
Marcel Datta	m.datta@tu-braunschweig.de
Marcel Warmbold	m.warmbold@tu-braunschweig.de
Stefan Sunken	s.sunken@tu-braunschweig.de

Braunschweig, 11. Mai 2016

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1	Lukas Schmidt	
2	Alexej Sivokha	
2.1	Alexej Sivokha	
2.2	Alexej Sivokha	
2.3	Alexej Sivokha	
2.4	Alexej Sivokha	
2.5	Alexej Sivokha	
3	Florian Rampold	
3.1	Florian Rampold	
3.2	Florian Rampold	
3.3	Florian Rampold	
4	Marcel Warmbold	

Inhaltsverzeichnis

1	Einleitung	6
2	Testplan	7
2.1	Zu testende Komponenten	7
2.2	Zu testende Funktionen/Merkmale	7
2.3	Nicht zu testende Funktionen	8
2.4	Vorgehen	8
2.4.1	Komponententests	9
2.4.2	Integrationstest	9
2.4.3	Systemtest	9
2.4.4	Abnahme und Funktionstests	9
2.5	Testumgebung	9
3	Abnahmetest	10
3.1	Zu testende Anforderungen	10
3.2	Testverfahren	12
3.2.1	Testskripte	12
3.3	Testfälle	12
3.3.1	Testfall <T100> - Post mit Text erstellen	13
3.3.2	Testfall <T200> - Post mit Foto erstellen	14
3.3.3	Testfall <T300> - Standorttest auf der Startseite	15
3.3.4	Testfall <T400> - Standort auf Post anzeigen	16
3.3.5	Testfall <T500> - Posts bewerten	17
3.3.6	Testfall <T600> - Kommentare erstellen	18
3.3.7	Testfall <T700> - Zeitstempel für Posts	19
3.3.8	Testfall <T800> - Posts sortieren	20
3.3.9	Testfall <T900> - Nachrichtenaustausch	21
3.3.10	Testfall <T1000> - Anzeige für eigene Posts	22
3.3.11	Testfall <T1100> - Anzahl der Kommentare	23
3.3.12	Testfall <T1200> - Datenbank	24
3.3.13	Testfall <T1300> - Posts löschen	25
3.3.14	Testfall <T1400> - Kommentare bewerten	26
3.3.15	Testfall <T1500> - Zeitstempel für Kommentar	27

3.3.16 Testfall $\langle T1600 \rangle$ - Standort für Kommentar	28
4 Glossar	29

Abbildungsverzeichnis

1 Einleitung

Um eine qualitativ hochwertige Software gewährleisten zu können ist es notwendig, die Software auf Erfüllung der für ihren Einsatz definierten Anforderungen zu prüfen und bewerten. Durch eine Reihe von Testfällen können Fehler behoben werden, welche sonst zu einer Beeinträchtigung des Produkts führen könnten. Jedoch kann nach den Tests nicht ausgeschlossen werden, dass die Software frei von Fehlern ist.

Die Android-Applikation Delay-Tolerant Jodel erlaubt Studierenden anonym den neusten Klatsch und Tratsch auszutauschen. Die Applikation bietet eine Art standortbezogenes Forum mit Voting- und Kommentarfunktion, um über die neuesten Infos oder Fotos zu diskutieren. Für die Kommunikation der Geräte über die App, liegt das Peer-to-Peer-Modell mithilfe des DTN-Bundles vor. Diese Software gilt es auf ihre wichtigsten Qualitätsmerkmale hin zu überprüfen.

Das Ziel der definierten Testfälle ist es, die Software in Bezug auf Funktionalität, Effizienz, Zuverlässigkeit, Benutzbarkeit, Veränderbarkeit und Übertragbarkeit (ISO/IEC 9126) zu testen, um die Softwarequalität sicherzustellen. Zudem soll sichergestellt werden, dass das Produkt anwenderfreundlich ist, sodass die App möglichst intuitiv zu bedienen ist und somit ein Einstieg ohne eine Einführung möglich ist.

2 Testplan

Im folgenden Abschnitt wird der Testplan des Softwaresystems dargestellt. Es wird auf die einzelnen Komponenten und Funktionen eingegangen, aus der die Software besteht. Außerdem wird die Vorgehensweise der einzelnen Tests beschrieben.

2.1 Zu testende Komponenten

In diesem Dokument werden folgende Komponenten für Android-Applikation getestet und analysiert:

- **Kamera**
Verwendet Kamera des Geräts für die Herstellung eines Fotos.
- **Standortdaten**
Verwendet das GPS-Modul von dem Gerät für die Erkennung des Standorts des Geräts.
- **Datenbank**
Speichert die Nachrichten in der Datenbank und bekommt sie aus der Datenbank.
- **Kommunikation**
Übertragung einer Nachricht (Post) zwischen den Geräten mit Hilfe von dem Bündelprotokoll, das unter Nutzung eines Delay-Tolerant Networks einen Austausch von Nachrichten Peer-to-Peer ermöglicht.

2.2 Zu testende Funktionen/Merkmale

Folgende Funktionen/Merkmale sind durch die Testverfahren zu testen.

- **<F10>** Post mit Text erstellen
- **<F20>** Post mit Foto erstellen
- **<F30>** Standort auf der Startseite
- **<F40>** Standort auf dem Post und Kommentar

- <F50> Post voten
- <F60> Post kommentieren
- <F70> Zeit für Posts und Kommentare richtig anzeigen
- <F80> Ansicht, in der Posts nach ihrem Voting sortiert angezeigt werden
- <F90> Nachrichtenaustausch funktioniert
- <F100> Post von Benutzer zeigen
- <F110> Kommentare den Posts zuordnen
- <F120> Kommentare bewerten
- <F130> Verbindung zur Datenbank funktioniert

2.3 Nicht zu testende Funktionen

Wir werden die zugrunde liegende Software, die zum Betrieb und Erstellen unserer Software dient, nicht testen. Dazu gehört z.B. das Betriebssystem auf der unsere Software läuft und die Tools mit der wir unsere Software geschrieben und kompiliert haben. Außerdem werden wir Softwarebibliotheken von Dritten nicht testen, ihre Korrektheit wird vorausgesetzt.

- Google Android
- Google Android SDK
- Google Android Studio
- IBR-DTN
- SQLite

2.4 Vorgehen

Im folgenden Abschnitt wird das Vorgehen für die einzelnen Testverfahren beschrieben, dabei wird das Vorgehen und die Art der Durchführung der Tests detailliert beschrieben.

2.4.1 Komponententests

Hier werden die Komponenten der Android Applikation isoliert voneinander mittels Testfällen getestet. Tester testen die Komponenten, wie es in den Testfällen beschrieben ist, und protokollieren ihre Ergebnisse schriftlich. Kritische Funktionalitäten der Android Applikation werden von Testskripten getestet. So ist nach erfolgreichem Abschluss der Tests die unabhängige Funktionalität einer Komponente sichergestellt.

2.4.2 Integrationstest

Nach erfolgreichen Komponententests folgt die Integration der einzelnen Komponenten mittels Bottom-Up-Prinzip. Nach dem zusammenführen einzelner Komponenten wird die Funktionalität von Testpersonen überprüft, die nach festgelegten Vorgehensweisen die Applikation testen. Ihre Beobachtungen werden schriftlich protokolliert.

2.4.3 Systemtest

Während des Systemtests wird die vollständige Erfüllung der funktionalen, sowie der nichtfunktionalen Anforderungen getestet. Es wird ein realistischer Betrieb der Software simuliert.

2.4.4 Abnahme und Funktionstests

Die Testfälle des Abnahmetests sollen gemeinsam mit dem Auftraggeber durchgeführt werden. Hierzu erhält der Auftraggeber vom Auftragnehmer vor Beginn der Abnahmetests eine Testanleitung, die beschreibt wie die im Pflichtenheft festgelegten Funktionen auf ihre Umsetzung hin geprüft werden können. Hierbei sollen insbesondere alle beschriebenen Anwendungsfälle mindestens einmal ausgeführt werden.

2.5 Testumgebung

Für die Komponententests und Integrationstests werden auf Seiten der App JUnit und die Android Test Suites genutzt. Abnahmetests benötigt man mindestens zwei Android Smartphones, die Android-Version 5.0 (Lollipop) unterstützen und mit Kamera, GPS-Modul und Internetverbindung ausgestattet sind.

3 Abnahmetest

In diesem Abschnitt werden die grundlegenden zu testenden Anforderungen an das Produkt definiert. Der Kunde muss das Produkt auf Vollständigkeit prüfen können. So wird nachgewiesen, dass die zwischen dem Auftraggeber und Auftragnehmer vereinbarten Leistungen erbracht wurden. Im folgenden Abschnitt soll insbesondere der Rahmen für den Test der Benutzeroberfläche und der Produktfunktionen festgelegt werden.

3.1 Zu testende Anforderungen

In diesem Abschnitt werden alle zu testenden Anforderungen aufgeführt

Nr	Anforderung	Testfälle	Kommentar
1	<F10> Können Posts mit Text erstellt werden?	<T100>	
2	<F20> Können Posts mit Foto erstellt werden?	<T200>	
3	<F30> Wird der Standort auf der Startseite korrekt angezeigt?	<T300>	
4	<F40> Wird der Standort auf dem Post angezeigt?	<T400>, <T1200>	
5	<F50> Kann der Benutzer Posts voten?	<T100>, <T200>, <T500>, <T1200>	
6	<F60> Kann der Benutzer Posts kommentieren?	<T600>, <T1200>, <T100>, <T200>	

7	<F70> Wird die Zeit, zu der ein Post erstellt wurde richtig angezeigt?	<T700>, <T1200>, <T100>, <T200>, <T600>	
8	<F80> Gibt es eine funktionierende Ansicht, in der Posts nach ihrem Voting sortiert angezeigt werden?	<T800>, <T1200>, <T100>, <T200>	
9	<F90> Funktioniert der Nachrichtenaustausch? Das heißt, bekommen Benutzer in unmittelbarer Nähe, die im selben WLAN sind, gegenseitig Posts, Kommentare und Votes übermittelt?	<T100>, <T200>,<T500> <T600>, <T900>, <T1200>	
10	<F100> Sieht der Benutzer seine selbsterstellten Posts in einer eigenen Ansicht?	<T100>, <T200>, <T1000>, <T1200>	
11	<F110> Wird die Anzahl der Kommentare zu den Posts richtig angezeigt?	<T100>, <T200>, <T600>, <T1100>, <T1200>	
12	<F120> Funktioniert die Verbindung zur Datenbank?	<T1200>	
13	<F120> Werden mehr als 150 Posts angezeigt?	<T100>, <T200>, <T1300>, <T1200>	
14	<F130> Können Kommentare bewertet werden?	<T100>, <T200>, <T600>, <T1400>, <T1200>	
15	<F70> Wird die Zeit, zu der ein Kommentar erstellt wurde richtig angezeigt?	<T1200>, <T100>, <T200>, <T600>,<T1500>	

16	<F130> Wird der Standort auf dem Kommentar angezeigt?	<T100>, <T200>, <T600>, <T1600>, <T1200>	
----	---	--	--

3.2 Testverfahren

Bei dem Testverfahren handelt es sich um einen Black-Box-Test. Hierbei finden Tests der Software statt, bei denen keine Kenntnisse über die innere Funktion des zu testenden Systems benötigt werden. Die Funktionsweise der Software wird überprüft, indem getestet wird, ob das Programm zu einer Eingabe die entsprechende Ausgabe anzeigt. Durch frühzeitiges Festlegen, wie die Software auf bestimmte Eingaben reagieren sollte, kann durch das Black Box Verfahren abgeglichen werden, ob die funktionalen Produktanforderungen erfüllt sind. Wichtig ist es, dabei auch Grenzfälle zu betrachten für die Eingaben gerade so noch akzeptiert werden oder gerade so abgelehnt werden.

3.2.1 Testskripte

Es werden keine Testskripte verwendet.

3.3 Testfälle

In diesem Abschnitt werden alle Testfälle für die App definiert.

3.3.1 Testfall $\langle T100 \rangle$ - Post mit Text erstellen

Ziel

Das Ziel ist es zu zeigen, dass Posts eines Benutzers auf der Startseite der App erscheinen. Somit wird sichergestellt, dass der Benutzer Posts auf seinem eigenen Endgerät sieht.

Objekte/Methoden/Funktionen

Objekte: Die App

Funktionen: **F10**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn der eingegebene Text in seinem Container(Cardview) auf der Startseite der App des Geräts erscheint. Der Vorgang schlägt fehl, wenn die App abstürzt oder die Eingabe nicht, beziehungsweise falsch entgegengenommen wird und nicht auf der Startseite dargestellt wird.

Vorbedingung

Die App muss ausgeführt werden und der Benutzer hat sich auf die Seite zum Erstellen eines Posts navigiert.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Plus-Button betätigen
3. Die Seite zum Erstellen eines Posts öffnet sich
4. Text wird in Textfeld eingegeben
5. Der Benutzer bestätigt mit Senden

Ausgabe: Der Benutzer gelangt zurück zur Startseite. Die Eingabe erscheint oben in der Ansicht.

Beobachtungen / Log / Umgebung

Beobachtet wird der Display auf der Startseite der App des Android Gerätes.

Besonderheiten

Bei diesem Test gibt es keine Besonderheiten.

Abhängigkeiten

Der Testfall hängt von Testfall $\langle T1200 \rangle$ (Verbindung zur Datenbank) ab.

3.3.2 Testfall $\langle T200 \rangle$ - Post mit Foto erstellen

Ziel

Das Ziel ist es zu zeigen, dass Fotoaufnahmen eines Benutzers auf der Startseite der App erscheinen.

Objekte/Methoden/Funktionen

Objekte: Die App

Funktionen: **F20**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn das Foto in seinem Container (Cardview) auf der Startseite der App des Geräts erscheint. Der Vorgang schlägt fehl, wenn die App abstürzt, das Foto nicht komprimiert werden kann oder die App nicht auf die Kamera des Benutzers zugreifen kann.

Vorbedingung Die App muss gestartet sein. Außerdem muss der Benutzer, wenn er über Android 6.0 oder höher verfügt den Zugriff auf seine Kamera erlauben. Außerdem muss der Benutzer auf die Seite zum Erstellen eines Posts navigieren können und den Kamera-Button drücken.

Eingabe:

Einzelschritte

1. Die App öffnen
2. Plus-Button betätigen
3. Die Seite zum Erstellen eines Posts öffnet sich
4. Der Benutzer betätigt den Kamera-Button
5. Der Benutzer schießt ein Foto
6. Das Foto wird in der Vorschau angezeigt
7. Der Benutzer bestätigt mit Senden

Ausgabe: Der Benutzer gelangt zurück zur Startseite. Das Foto erscheint in seinem Container oben in der Ansicht.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite der App des Android Gerätes.

Besonderheiten

Bei diesem Test gibt es keine Besonderheiten.

Abhängigkeiten

Der Testfall hängt von Testfall $\langle T1200 \rangle$ (Verbindung zur Datenbank) ab.

3.3.3 Testfall $\langle T300 \rangle$ - Standorttest auf der Startseite

Ziel

Überprüfung des Normalfalls: Dem Benutzer werden mittels GPS seine aktuellen Lokationsdaten angezeigt.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation und GPS-Modul ausgeführt

Funktionen: **F30**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn der Benutzer auf der Startseite seinen aktuellen Standort sieht. Außerdem muss der Standort wechseln, wenn der Benutzer seine Lokationsdaten ändert.

Vorbedingung Die App muss gestartet sein. Das Gerät muss über ein GPS-Modul verfügen und dieses muss aktiviert sein.

Einzelschritte

Eingabe:

1. Die App öffnen

Ausgabe: Der Standort wird auf der Startseite angezeigt.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite der App des Android Gerätes.

Besonderheiten

Der Standort kann nur angezeigt werden, wenn die GPS-Funktion aktiviert ist.

Abhängigkeiten

Es gibt keine Abhängigkeiten

3.3.4 Testfall $\langle T400 \rangle$ - Standort auf Post anzeigen

Ziel

Überprüfung des Normalfalls: Jeder Post enthält eine Anzeige, die darstellt von wo aus der Post abgesendet wurde.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation und GPS-Modul ausgeführt

Funktionen: **F40**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn der Benutzer auf der Startseite und der Detailseite für Kommentare zu einem Post den Standort sieht, von dem dieser abgeschickt wurde. Der Test gilt als fehlgeschlagen, wenn kein Standort angezeigt wird oder der Standort im Nachhinein wechselt.

Vorbedingung Die App muss gestartet sein. Das Gerät muss über ein GPS-Modul verfügen und dieses muss aktiviert sein.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Post verfassen

Ausgabe: Der Standort wird auf dem Container für Posts angezeigt.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite und der Seite für Kommentare der App des Android Gerätes.

Besonderheiten

Diese Funktion kann erst getestet werden, wenn ein Post erstellt wurde.

Abhängigkeiten

Der Testfall hängt von Testfall $\langle T1200 \rangle$ (Verbindung zur Datenbank), $\langle T100 \rangle$, $\langle T200 \rangle$ ab.

3.3.5 Testfall $\langle T500 \rangle$ - Posts bewerten

Ziel

Überprüfung des Normalfalls: Die Votinganzeige eines Posts oder Kommentars ändert sich, wenn der Benutzer den Upvote- oder Downvotebutton betätigt.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation und Datenbank ausgeführt.

Funktionen: **F50**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn sich die Anzeige der Votes nach dem Voten eines Posts oder eines Kommentars um einen Wert verändert und dauerhaft gespeichert wird. Fehlgeschlagen gilt die Aktion, wenn die Anzeige gleichbleibt oder die Anzeige ihren Wert verändert nachdem der Benutzer durch verschiedene Seiten der App navigiert.

Vorbedingung Die App muss gestartet sein. Eine Verbindung zur Datenbank muss bestehen.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Auf Startseite oder Detailseite für Kommentare navigieren
3. Upvote- oder Downvotebutton betätigen

Ausgabe: Die Anzeige verändert sich um einen Wert nach oben beziehungsweise nach unten.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite und der Seite für Kommentare der App des Android Gerätes.

Besonderheiten

Diese Funktion kann erst getestet werden, wenn ein Post beziehungsweise ein Kommentar erstellt wurde.

Abhängigkeiten

Der Testfall hängt von $\langle T1200 \rangle$ (Verbindung zur Datenbank), $\langle T100 \rangle$ und $\langle T200 \rangle$ ab.

3.3.6 Testfall <T600> - Kommentare erstellen

Ziel

Überprüfung des Normalfalls: Nachdem die Seite zum Kommentieren von Posts aufgerufen wurde und ein Text eingegeben und abgesendet wird, erscheint der Kommentar zu dem zugehörigen Post auf der Kommentarseite.

Objekte/Methoden/Funktionen

Objekte: Hierbei wird die Komponente Applikation ausgeführt.

Funktionen: **F60**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn der eingegebene Kommentar auf der Kommentarseite zu dem zugehörigen Post erscheint.

Vorbedingung Die App muss gestartet sein. Die Seite zum Erstellen eines Kommentars ist geöffnet.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Auf Post drücken, der kommentiert werden soll
3. Kommentieren-Button betätigen
4. Seite zum Kommentare erstellen öffnet sich
5. Text wird eingegeben
6. Senden-Button wird gedrückt

Ausgabe: Der Kommentar erscheint auf der Kommentarseite und wird unter den bereits bestehenden Kommentaren zu dem zugehörigen Post angezeigt.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Seite für Kommentare der App des Android Gerätes.

Besonderheiten

Diese Funktion kann erst getestet werden, wenn ein Post erstellt wurde.

Abhängigkeiten

Der Testfall hängt von <T1200> (Verbindung zur Datenbank) und Testfall <T100>, sowie <T200> ab.

3.3.7 Testfall <T700> - Zeitstempel für Posts

Ziel

Überprüfung des Normalfalls: Die Zeit zu der ein Post erstellt wurde, wird auf dem Container für diese angezeigt.

Objekte/Methoden/Funktionen

Objekte: Hierbei wird die Komponente Applikation ausgeführt.

Funktionen: **F70**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn nach Erstellen eines Posts die richtige Zeit angezeigt wird.

Vorbedingung Die App muss gestartet sein.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Post erstellen

Ausgabe: Die Zeit zu der der Post oder Kommentar abgesendet wurde, wird auf dem Container angezeigt.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite der App des Android Gerätes.

Besonderheiten

Bei diesem Test gibt es keine Besonderheiten.

Abhängigkeiten

Der Testfall hängt von <T1200> (Verbindung zur Datenbank) und <T100>, sowie <T200> ab.

3.3.8 Testfall $\langle T800 \rangle$ - Posts sortieren

Ziel

Überprüfung des Normalfalls: Nach dem Betätigen einer Schaltfläche werden die Posts absteigend nach der Anzahl ihrer Votes sortiert angezeigt.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation und Datenbank ausgeführt.

Funktionen: **F80**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn die Posts in der Datenbank gespeichert sind und absteigend sortiert angezeigt werden.

Vorbedingung Die App muss gestartet sein. Es existieren bereits mehrere Posts

Einzelschritte

Eingabe:

1. Die App öffnen
2. Auf Schaltfläche für das Sortieren von Posts drücken

Ausgabe: Es erscheint eine neue Seite auf der die Posts nach Votes absteigend sortiert werden.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite der App des Android Gerätes.

Besonderheiten

Bei diesem Test gibt es keine Besonderheiten.

Abhängigkeiten

Der Testfall hängt von $\langle T1200 \rangle$ (Verbindung zur Datenbank) und $\langle T100 \rangle$ beziehungsweise $\langle T200 \rangle$ ab.

3.3.9 Testfall $\langle T900 \rangle$ - Nachrichtenaustausch

Ziel

Überprüfung des Normalfalls: Sowohl Posts, als auch Kommentare und Votings werden an alle Benutzer der App weitergeleitet, die sich im selben WLAN und Reichweite eines Benutzers befinden, der die Nachricht bereits empfangen hat.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation, Bündelprotokoll und Datenbank ausgeführt.

Funktionen: **F90**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn der abgesendete Post Kommentar und Vote mithilfe des Bündelprotokolls, wie in Ziele beschrieben übermittelt wird. Der Test schlägt fehl, wenn Benutzer, die sich im selben WLAN befinden, von der die Nachricht ausgeht, den Post, Kommentar oder Vote nicht sehen können.

Vorbedingung Die App muss gestartet sein. Das Bündelprotokoll IBR DTN muss implementiert sein und eine Verbindung zur Datenbank muss bestehen.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Post verfassen
3. Für diesen Post einen Kommentar verfassen
4. Sowohl Post als auch Kommentar bewerten

Ausgabe: Der Post, Kommentar und Vote werden Benutzern, die sich im selben WLAN befinden übermittelt.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite, sowie der Seite für Kommentare der App des Android Gerätes.

Besonderheiten

Bei diesem Test gibt es keine Besonderheiten.

Abhängigkeiten

Der Testfall hängt von $\langle T1200 \rangle$ (Verbindung zur Datenbank), $\langle T100 \rangle$, $\langle T200 \rangle$, $\langle T500 \rangle$ und $\langle T600 \rangle$ ab.

3.3.10 Testfall $\langle T1000 \rangle$ - Anzeige für eigene Posts

Ziel

Überprüfung des Normalfalls: Durch Navigieren auf eine Schaltfläche, wird der Benutzer auf eine Seite weitergeleitet, auf der er seine selbstverfassten Posts sehen kann.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation und Datenbank ausgeführt

Funktionen: **F100**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn der Benutzer seine eigenen Posts sehen kann.

Vorbedingung Die App muss gestartet sein und eine Verbindung zur Datenbank muss existieren.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Auf Button drücken, mit dem eigene Posts angezeigt werden

Ausgabe: Die Seite für eigene Posts öffnet sich

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Seite für die eigenen Posts der App des Android Gerätes.

Besonderheiten

Der Benutzer muss vorher selber einen Post verfasst haben.

Abhängigkeiten

Der Testfall hängt von $\langle T1200 \rangle$ (Verbindung zur Datenbank) und $\langle T100 \rangle$ beziehungsweise $\langle T200 \rangle$ ab.

3.3.11 Testfall $\langle T1100 \rangle$ - Anzahl der Kommentare

Ziel

Überprüfung des Normalfalls: Kommentare werden zu den zugehörigen Posts gespeichert und angezeigt.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation und Datenbank ausgeführt.

Funktionen: **F110**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn sich die Anzeige für die Anzahl der Kommentare auf dem Container für Posts erhöht, nachdem einem Post ein Kommentar hinzugefügt wurde.

Vorbedingung Die App muss gestartet sein. Es muss bereits mindestens ein Post existieren, der kommentiert werden kann.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Auf Post drücken, der kommentiert werden soll
3. Kommentar verfassen

Ausgabe: Die Anzahl der Kommentare des Posts hat sich auf dem jeweiligen Container für diesen um einen Wert verändert.

Beobachtungen / Log / Umgebung Beobachtet wird der Container der Posts auf der Startseite der App des Android Gerätes.

Besonderheiten

Es muss bereits ein Post erstellt worden sein.

Abhängigkeiten

Der Testfall hängt von $\langle T1200 \rangle$ (Verbindung zur Datenbank) und $\langle T100 \rangle$ beziehungsweise $\langle T200 \rangle$ und $\langle T600 \rangle$ ab.

3.3.12 Testfall $\langle T1200 \rangle$ - Datenbank

Ziel

Überprüfung des Normalfalls: Die Datenbank ist funktionstüchtig und speichert Votes, Posts, Kommentare, den Standort und den Zeitstempel einer Nachricht.

Objekte/Methoden/Funktionen

Objekte: Hierbei wird die Komponente Datenbank ausgeführt

Funktionen: **F130**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn die Testfälle $\langle T100 \rangle$, $\langle T200 \rangle$, $\langle T300 \rangle$, $\langle T400 \rangle$, $\langle T500 \rangle$, $\langle T600 \rangle$, $\langle T700 \rangle$, $\langle T800 \rangle$, $\langle T900 \rangle$, $\langle T1000 \rangle$, $\langle T1100 \rangle$, $\langle T1300 \rangle$ erfolgreich sind.

Vorbedingung Die App muss gestartet sein. Alle anderen aufgeführten Testfälle müssen durchgeführt werden.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Testfälle 100 - 1100 und 1300 und 1400 ausführen.

Ausgabe: Alle Testfälle laufen wie erwartet ab.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite, sowie der Seite für Kommentare der App des Android Gerätes.

Besonderheiten

Bei diesem Test gibt es keine Besonderheiten.

Abhängigkeiten

Bei diesem Test gibt es keine Abhängigkeiten.

3.3.13 Testfall <T1300> - Posts löschen

Ziel

Überprüfung des Normalfalls: Die Datenbank löscht Posts, wenn mehr als 150 Posts empfangen wurden.

Objekte/Methoden/Funktionen

Objekte: Hierbei wird die Komponente Datenbank und App ausgeführt

Funktionen: **F130**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn die Anzahl der Posts von 150 nicht überschritten wird.

Vorbedingung Die App muss gestartet sein. Es müssen mindestens 151 Posts empfangen worden sein.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Mehr als 150 Posts erstellen

Ausgabe: Es dürfen nicht mehr als 150 Posts auf der Startseite angezeigt werden.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Startseite der App des Android Gerätes.

Besonderheiten

Bei diesem Test gibt es keine Besonderheiten.

Abhängigkeiten

Der Testfall hängt von Testfall <T1200> sowie <T100> und <T200> ab.

3.3.14 Testfall $\langle T1400 \rangle$ - Kommentare bewerten

Ziel

Überprüfung des Normalfalls: Die Votinganzeige eines Kommentars ändert sich, wenn der Benutzer den Upvote- oder Downvotebutton betätigt.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation und Datenbank ausgeführt

Funktionen: **F120**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn sich die Anzeige der Votes nach dem Bewerten eines Kommentars um einen Wert erhöht und dauerhaft gespeichert wird. Fehlgeschlagen gilt die Aktion, wenn die Anzeige gleichbleibt oder die Anzeige ihren Wert verändert nachdem der Benutzer durch verschiedene Seiten der App navigiert.

Vorbedingung Die App muss gestartet sein. Eine Verbindung zur Datenbank muss bestehen.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Auf Startseite oder Detailseite für Kommentare navigieren
3. Upvote- oder Downvotebutton betätigen

Ausgabe: Die Anzeige verändert sich um einen Wert nach oben beziehungsweise nach unten.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Seite für Kommentare der App des Android Gerätes.

Besonderheiten

Diese Funktion kann erst getestet werden, wenn ein Kommentar erstellt wurde.

Abhängigkeiten

Der Testfall hängt von $\langle T1200 \rangle$ (Verbindung zur Datenbank) und $\langle T100 \rangle$ beziehungsweise $\langle T200 \rangle$ und $\langle T600 \rangle$ ab.

3.3.15 Testfall <T1500> - Zeitstempel für Kommentar

Ziel

Überprüfung des Normalfalls: Die Zeit zu der ein Kommentar erstellt wurde, wird auf dem Container für diese angezeigt.

Objekte/Methoden/Funktionen

Objekte: Hierbei wird die Komponente Applikation ausgeführt.

Funktionen: **F70**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn nach Erstellen eines Kommentars die richtige Zeit angezeigt wird.

Vorbedingung Die App muss gestartet sein. Es wurde bereits ein Post und ein Kommentar dazu erstellt.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Kommentar erstellen

Ausgabe: Die Zeit zu der ein Kommentar abgesendet wurde, wird auf dem Container angezeigt.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Seite für Kommentare der App des Android Gerätes.

Besonderheiten

Bei diesem Test gibt es keine Besonderheiten.

Abhängigkeiten

Der Testfall hängt von <T1200> (Verbindung zur Datenbank) und <T100>, sowie <T200> und <T600> ab.

3.3.16 Testfall $\langle T1600 \rangle$ - Standort für Kommentar

Ziel

Überprüfung des Normalfalls: Jeder Kommentar enthält eine Anzeige, die darstellt von wo aus der Kommentar abgesendet wurde.

Objekte/Methoden/Funktionen

Objekte: Hierbei werden die Komponenten Applikation und GPS-Modul ausgeführt

Funktionen: **F40**

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn der Benutzer der Detailseite für Kommentare zu einem Post den Standort sieht, von dem dieser abgeschickt wurde. Der Test gilt als fehlgeschlagen, wenn kein Standort angezeigt wird oder der Standort im Nachhinein wechselt.

Vorbedingung Die App muss gestartet sein. Das Gerät muss über ein GPS-Modul verfügen und dieses muss aktiviert sein.

Einzelschritte

Eingabe:

1. Die App öffnen
2. Kommentar verfassen

Ausgabe: Der Standort wird auf der Kommentarseite auf dem Container für Kommentare angezeigt.

Beobachtungen / Log / Umgebung Beobachtet wird der Display auf der Seite für Kommentare der App des Android Gerätes.

Besonderheiten

Diese Funktion kann erst getestet werden, wenn ein Post beziehungsweise ein Kommentar erstellt wurde.

Abhängigkeiten

Der Testfall hängt von Testfall $\langle T1200 \rangle$, $\langle T100 \rangle$, $\langle T200 \rangle$ und $\langle T600 \rangle$ ab.

4 Glossar

IBR-DTN: Bündelprotokoll für DTN-Anwendungen entwickelt vom Institut für Betriebssysteme und Rechnerverbund der der TU-Braunschweig.

Post: engl. für Nachricht

Peer-to-Peer: Kommunikationsart (Nutzer-zu-Nutzer)

App: Abkürzung von Applikation, wird fast ausschließlich mit mobiler App gleichgesetzt. Sie stellt eine Anwendungssoftware, vorwiegend für den mobilen Gebrauch dar.

GPS: Global Positioning System ist ein globales Navigationssatellitensystem zur Positionsbestimmung.

Android-Studio: Eine Entwicklungsumgebung für Android.

Android SDK: Ein Software Development Kit (SDK) ist eine Sammlung fertiger Programme (sogen. Werkzeuge und Anwendungen), um ein neues Programm (Software) zu erstellen, meist inklusive Dokumentation.

SQLite: SQLite ist eine Programmbibliothek, die ein relationales Datenbanksystem enthält.