#9 User will einen Kontakt hinzufügen:

- Dem User soll es möglich sein jeden anderen User als Kontakt hinzuzufügen ohne deren Bestätigung.
- Dem User soll es nicht möglich sein denselben Kontakt mehrmals hinzu zu fügen.

Test: UserITCase – testAddUser()
Seite aufrufen – Anzahl feststellten der hinzufügbaren User und der bereits hinzugefügten User – einen User hinzufügen – Anzahl der hinzufügbaren muss sich um eins verringert haben – Anzahl hinzugefügter User muss sich um eins erhöht haben.

#10 User will im Header einen Link zum eigenen Profil

- Link muss im Header auf allen Seiten verfügbar sein.
- Im Header muss ein Profilbild und der Name angezeigt werden.
- Der User soll über einen Link zu seinem eigenen Profil kommen.

Test: UserITCase - testUserProfilePresent()
Aufrufen der Profile Seite und verifizieren von einigen User Daten.

Test: UserITCase – TestAccessUserProfileFromActivityStream() + communityOveriew() + CommunityProfile() + UserOverview()
Aufrufen der einzelnen Seiten und überprüfen ob der link vorhanden ist. Dabei wird von der jeweiligen Seite auf das Profil gesprungen und geprüft ob es tatsächlich die Profilseite ist.

#11 User will Kontakt entfernen

- Dem User soll es möglich sein jeglichen Kontakt zu entfernen.
- Eine Entfernung findet ohne Bestätigung des zu Entfernenden Kontakts statt.
- Einem entfernten User muss es möglich sein den Kontakt wieder hinzu zufügen.

Test: UserITCase – testAddUser()
UserOverViewPage wird aufgerufen. Alle Buttons „Add Contact“ und „Remove Contact“ werden gezählt. Der User mit Id 2 wird hinzugefügt. Wieder alle Buttons gezählt und verglichen. Add Contact muss sich um einen verringert Remove Contact um einen erhöht haben. Wieder entfernen des Users und erneute Überprüfung der Counter welche wieder auf den ursprünglichen Wert stehen müssen.

#12 Als Administrator will ich neu erstellte Communities freischalten.

- Als Administrator möchte ich alle Communities sehen welche auf eine Freischaltung warten.

Test: AdminITCase – testPendingCommunitiesListPresent()
Admin Seite wird geladen und die Pending Communities werden gezählt.

Test: AdminITCase – testApproveCommunity()
Admin Seite wird geladen und die erste Community wird auf Approved gesetzt. Es wird überprüft ob die Community nicht mehr in der Liste der Pending Communities ist jedoch in der Liste der Aprroved Communities.

#13 User will mittels Navigation im Header Zugriff zu alle relevanten Seiten haben

- Folgende Seiten müssen in der Navigation vorhanden sein: Home, Communities, Eigenes Profil
- User muss von jeder Seite über die Navigation zurückkommen.
- Die Navigation muss auf allen Seiten verfügbar sein.

Test: testUserListPresent()

Aufruf der UserOverviewPage über die einzelnen Seiten

Test: testCommuniyListPresent()
Aufruf der CommunityOverView über die einzelnen Seiten
Test: Weitere Tests werden immer wieder über verschiedene Wege durchgeführt.

#14 Als Portaladmin möchte ich Dokumente für User als Download zur Verfügung stellen.

- PDF, DOC, JPG, Dateien können zur Verfügung gestellt werden (Download)

Test: PortalAdminITCase – testUploadFile()
Erstellen einer temporären Datei – Wechsel auf community Seite – Upload einer jpg Datei – aktualisieren der Seite – Files auf der Seite in Liste laden – verifizieren ob die Datei vorhanden ist.

#16 Als Portaladmin möchte ich Dokumente editieren

- Textdokumente sollten editierbar sein
- ~~Konvertierung HTML in PDF soll möglich sein~~
- Bilder können bearbeitet werden

Test: testUploadFile()
Erstellen einer File. Hochladen der Datei und überprüfen ob dies funktioniert hat.

#17 Als user möchte ich einen zentralen Activity-Stream um eine gute Übersicht zu haben

- dient als Zeitverlauf
- Einträge sind leichter abgrenzbar vom Rest der Seite

Test: testMostRecentMessagesOnTop(), TestPostsPresent()
Sieh auch Userstory #29 #19

#19 User will Posts schreiben können

- Nachrichten können global für alle sichtbar verfasst werden.
- Nachrichten können auf Communities beschränkt werden, wenn User Teil der Community ist.
- Es wird keine API für Bots angeboten.
- Dem User soll auf der eigenen Profilseite ein Kommentarfeld angezeigt werden, indem er seine Nachrichten verfassen kann.
- Der User kann die Sichtbarkeit für jede Nachricht einzeln festlegen.
- Es muss ein Sende-Button zum Abschicken der Nachricht vorhanden sein.

Test: UserITCase – testNewPost()
Nachricht wird erstellt – Activity Stream aufgerufen – Post abgesetzt – aktualisieren (automatisch) – alle Posts auslesen und verifizieren das der neue enthalten ist.

#20 Als Portaladmin möchte ich News bearbeiten können.

- erstellen und löschen von Posts
- alle Rollen sollen Lese-Rechte haben

Test: PortalAdminITCase - testDeletePost()
Eine Nachricht wird erstellt und in den Newsbereich der Community des Portaladmins gepostet. Verifizieren des erstellten Posts. Löschen des Posts und verifizieren.

#21 User will Feedback auf Nachrichten in Activity Stream geben können

- Es muss einen Like-Button sowie ein Kommentarfeld geben.
- Es darf nur positive Rückmeldungen geben (= nur Like-Button).

- Feedback hat immer die gleiche Sichtbarkeit wie die Nachricht.
- Beim Feedback muss der Name des Users dabeistehen.

Test: UserITCase - testLikeFirstPost()
Erzeugen einer Nachricht. Verifizieren dass wir den ersten nicht liken, danach wird er gelikt und verifiziert ob wir angeführt werden. Danach wider unliked und verifiziert.

#22 Der User will den Newsbereich eines Unternehmens lesen können.

- Der Newsbereich wird durch Inhalte und später Fotos gestaltet.
- Admins sollen diese editieren können.
- Online Zeiten von/bis sollen angezeigt werden.

Test: siehe #29

#23 User will auf der Startseite nur globale Nachrichten und die seiner Communities sehen

- Der Activity Stream muss personalisiert auf den User sein.
- Es dürfen nur für den User relevante Nachrichten angezeigt werden (=global & Community).

Test: UserITCase - testOnlyGlobalAndMemberPosts()
Erstellen einer Liste der Communities derer der User nicht angehört. Auslesen Post Headers auf der ActivityStream Seite. Die AntiComs dürfen nicht in der Liste der Headers enthalten sein. Durch Ausschlussverfahren können also nur die anderen möglichen Communities und die Globalen Nachrichten vorhanden sein.

#26 User will eine Übersicht der Communities

- Die Communities sollen als Übersicht dargestellt werden
- Wenn man im Header auf Communities klickt, soll die Communitiesübersicht angezeigt werden -> ersetzt Activitystream
- Der User kommt durch den Klick auf die Community auf die entsprechende Seite mit Activity Stream.

Test: UserITCase - testOnlyCommunityMessagesOnCommunityProfilePage()
Community Seite wird aufgerufen und in jedem Post die Community ausgelesen. Verifizieren, dass die Community immer nur die der gewünschten Community ist.

#28Administrator kann neue Communities freischalten

- Administrator erhält eine Benachrichtigung (private Message) wenn eine neue Community beantragt wurde.
- Administrator kann die Anfrage ablehnen oder die Community freischalten.
- Administrator sieht ob die Community als "private" oder "öffentliche" Community beantragt wurde.

Test: AdminITCase - testDeclineCommunity()
Öffnen der Administrationsseite und ablehnen der ersten Community im Bereich Pending. Verifiziert wird, indem in der Liste der Pending Communities die rejected Community nicht mehr enthalten ist.

Test: AdminITCase – testApproveCommunity()
Freischalten der ersten Pending Community. Verifiziert wird, indem überprüft wird, ob die Community in den Approved Communities Bereich aufscheint.

#29 User soll in der Community den jeweiligen Activity Stream sehen

- Es sollen keine globalen Nachrichten sichtbar sein.
- Die aktuellsten Nachrichten sollen nach vorne gereiht werden.
- User muss Teil der Community sein um den Activity Stream zu sehen.

Test: UserITCase - testOnlyCommunityMessagesOnCommunityProfilePage()
Siehe #26

Test: UserITCase - testUserAccessCommunitiesPage()
User ruft eine Community in der er Mitglied ist auf. Es wird verifiziert, dass Leave und nicht Join Button vorhanden ist. Verifizieren ob Posts vorhanden sind.

#32 Als user möchte ich auf meiner Userseite Abteilungen und Kontake einsehen können.

- Liste Abteilungen (Community)
- Liste Kontakte

Test: UserITCase - testDepartmentsAndContactsPresent()
Aufrufen der eigenen Profilseite. Auslesen der Contacts- und Communities- Anzahl. Diese muss größer 0 sein.

#33User kann eine Community als öffentlich oder privat beantragen

- Beantragung muss durch Administrator freigegeben werden.
- Jeder angemeldete User kann eine Beantragung starten.
- User erhält eine Bestätigungs-Mail.

Test: UserITCase - testCreatePublicCommunity()
Erstellen einer Community als User. Danach wird als Admin die Pending Communities ausgelesen und der Inhalt auf die erstellten Communities überprüft

Test: UserITCase – testCreateprivateCommunity()
Gleich wie oben nur dass diese private gesetzt ist.

#34 Als User möchte ich Userinformationen haben

- Informationen beinhalten Vorname, Nachname, E-Mail Adresse (=username) und Passwort
- Informationen müssen geschäftliche Inhalte (Abteilung/Team, Geographische Adresse/Ort und Zimmer, Telefonnummer) erfüllen
- Angabe von Kontakten, Expertisen und Interessen soll möglich sein
- User kann Profilbild hochladen

Test: UserITCase - testUserProfilePresent()
Aufrufen der Profilseite und auslesen von Vorname, Nachname, Username und E-Mail Adresse

#37 Als User möchte ich die öffentlichen Activities meiner Kontakte sehen

- Verifizieren, dass User nur öffentliche Activities der eigenen Kontakte sehen kann.
- Verifizieren, dass User die Activities von ihren Kontakten nicht bearbeiten können.

Test: UserITCase - testOnlyGlobalAndMemberPosts()
Punkt Eins wird mit der Userstorie #23 verifiziert.

Punkt Zwei wird bzw. kann nicht verifiziert werden, da dies voraussetzt, dass der User Post bearbeitet werden kann, was derzeit nicht möglich ist. Dazu hätte #15 umgesetzt werden müssen.