

Testdokumentation

Antrags- und Beschlussverwaltungstool

Inhaltsverzeichnis

1. Teststrategie	1
2. Testkonzept	2
3. Testfallbeschreibung	3
3.1. Vorbedingungen	3
3.2. Test Cases	3
3.3. TC01 [Initialisierung] - Erfolgreich Initialisiert	3
3.4. TC02 [Login] - Gültige Eingaben akzeptiert	3
3.5. TC03 [Login] - Ungültige Eingaben abgelehnt	3
3.6. TC04 [Antrag] - Gültigen Antrag einreichen	4
3.7. TC05 [Antrag] - Anhänge werden überliefert	4
3.8. TC06 [Antrag] - Ungültigen Antrag ablehnen	4
3.9. TC07 [Antrag] - Antrag anzeigen	5
3.10. TC08 [Antrag] - Antrag bearbeiten	5
3.11. TC09 [Tagesordnung] - Erfolgreiches Generieren der Tagesordnung	5
3.12. TC10 [Sitzung] - Sitzung vertagen	6
3.13. TC11 [Antrag] - Antrag einer anderen Sitzung zuweisen	6
3.14. TC12 [DB] - Verlust der Verbindung zur DB	7
3.15. TC13 [Referat] - neues Referat erstellen	7
3.16. TC14 [Referat] - Referat Löschen	7
3.17. TC15 [Sitzung] - Sitzung erstellen	7
3.18. TC16 [Sitzung] - Tagesordnung/Sitzung abschließen	8
3.19. TC17 [Sitzung] - Tagesordnung/Sitzung mit nicht abgeschlossenen Anträgen abschließen	8
3.20. TC18 [Sitzung] - in Zukunft liegende Sitzung abschließen	9
4. Konsequenzen	10

Kapitel 1. Teststrategie

Unser Fokus beim Testen lag auf dem Stellen von Anträgen sowie der Generierung der Tagesordnung in Verbindung mit einer Sitzung, da diese die Grundfunktionalitäten abbilden.

Zudem handelt es sich um die meistgenutzten Funktionen, bei denen jedoch auch die meisten Fehler auftreten können, da es zu Inkonsistenzen bei den Anträgen kommen kann.

Des Weiteren haben wir auch Tests der Datenbank sowie Login und Logout priorisiert, da der Stura besonderen Wert auf Konsistenz und Authentifizierung legt.

Kapitel 2. Testkonzept

Es wurden die Komponenten Anträge, Referate, Sitzungen und Login getestet. Die ersten Testinstanzen waren manuelle Tests, die vom Entwickler während der Entwicklung oder danach durchgeführt wurden. Dabei wurden diese Komponenten einzeln auf ihre Funktionalitäten und Methoden überprüft, mithilfe von Unittests. Es wurden Unittests für die Datenbankinitialisierung, den Login, Logout und die Antragstellung implementiert.

Nachdem die Komponententests erfolgreich waren, wurden auch Integrationstests durchgeführt, wie zum Beispiel die Kombination von Login und Anträgen sowie von Anträgen und Sitzungen bis zur Generierung von Tagesordnungen.

Anschließend wurden Smoke-Tests und End-to-End-Tests von Teammitgliedern durchgeführt. In Bezug auf die Testdurchführungsplanung wurde der jeweilige Test-Verantwortliche beauftragt, einen Test für eine implementierte Funktionalität zu erstellen, sobald diese vom Entwickler freigegeben wurde. Anschließend wurde der Test durchgeführt.

Kapitel 3. Testfallbeschreibung

3.1. Vorbedingungen

Um die Software nutzen zu können, müssen die folgenden Bedingungen gelten:

1. Es besteht eine aktive Internetverbindung
2. Der Benutzer befindet sich im internen Netzwerk der HTW Dresden
3. Der Host-Server ist online und erreichbar

3.2. Test Cases

3.3. TC01 [Initialisierung] - Erfolgreich Initialisiert

Beschreibung	Das System wird gestartet und das Datenbankmodell wird erfolgreich initialisiert. [Test Automatisiert]
Vorbedingung	Das System ist startbereit auf dem Server eingerichtet.
Nachbedingung	Modelldaten sind in der Datenbank gespeichert.
Benötigte Daten	Antragsdaten, Referatsdaten, Sitzungsdaten
Testergebnis	Die Initialisierung des Systems und die Speicherung der Modelldaten in der Datenbank sind erfolgreich.
Konsequenz	Das Thema "Datenbankinitialisierung" wird abgeschlossen, und Ressourcen zur Bearbeitung von anderen Funktionen werden freigegeben.

3.4. TC02 [Login] - Gültige Eingaben akzeptiert

Beschreibung	Der Benutzer gibt gültige Logindaten ein.
Vorbedingung	Dem Benutzer wurde bereits ein Login angelegt.
Nachbedingung	Zeitstempel und Benutzerkennung müssen in ein Log gespeichert werden. Der Login-Vorgang muss erfolgreich beendet werden.
Benötigte Daten	Gültige Logindaten (Benutzername, Passwort)
Testergebnis	Der Login-Vorgang akzeptiert gültige Logindaten und speichert Zeitstempel und Benutzerkennung in einem Log.
Konsequenz	Login mit gültigen Logindaten funktioniert.

3.5. TC03 [Login] - Ungültige Eingaben abgelehnt

Beschreibung	Der Benutzer gibt ungültige Logindaten ein. Die Software muss den Login-Vorgang abbrechen.
---------------------	---

Vorbedingung	Es existiert ein gültiger Login.
Nachbedingung	Ausgabe einer Fehlermeldung.
Benötigte Daten	Logindaten (Benutzername, Passwort)
Testergebnis	Der Login-Vorgang lehnt ungültige Logindaten ab und gibt eine Fehlermeldung aus.
Konsequenz	Prüfung von ungültigen Logins funktioniert und es kommt eine entsprechende Fehlermeldung.

3.6. TC04 [Antrag] - Gültigen Antrag einreichen

Beschreibung	Der Benutzer hat einen Antrag vollständig ausgefüllt und abgeschickt. Der Antrag muss an die Datenbank gesendet und der Vorgang beendet werden.
Vorbedingung	Es existiert mindestens ein Referat und eine Sitzung.
Nachbedingung	Der Antrag muss in die Datenbank eingefügt und der korrekten nächsten Sitzung des ausgewählten Referats zugewiesen werden.
Benötigte Daten	Antragsdaten
Testergebnis	Ein vollständig ausgefüllter Antrag wird erfolgreich an die Datenbank gesendet und der nächstfolgenden Sitzung des ausgewählten Referats zugewiesen.
Konsequenz	Das Stellen eines Antrages mit gültigen Daten funktioniert.

3.7. TC05 [Antrag] - Anhänge werden überliefert

Beschreibung	Der Benutzer hat einen Antrag vollständig ausgefüllt und mit Anhängen (1x PDF, 1x DOCX & 2x PNG) abgeschickt.
Vorbedingung	Es existiert mindestens ein Referat und eine Sitzung.
Nachbedingung	Der Antrag muss in die Datenbank eingefügt und die Anhänge in einer separaten Tabelle, welche auf den Antrag verweist, abgespeichert werden.
Benötigte Daten	Antragsdaten, Anhänge
Testergebnis	Ein Antrag mit Anhängen (PDF, DOCX und PNG) wird in die Datenbank eingefügt und die Anhänge werden in einer separaten Tabelle abgespeichert.
Konsequenz	Anhänge werden erfolgreich abgespeichert.

3.8. TC06 [Antrag] - Ungültigen Antrag ablehnen

Beschreibung	Der Benutzer hat einen unvollständigen Antrag abgeschickt. Der Vorgang muss abgebrochen werden.
Vorbedingung	Es existiert mindestens ein Referat und eine Sitzung.
Nachbedingung	Es wird eine Fehlermeldung ausgegeben, die dem Benutzer aufzeigt, welche Angaben fehlen/ungültig sind.

Benötigte Daten	Antragsdaten
Testergebnis	Der Antrag wird nicht abgeschickt und es wird eine Fehlermeldung angezeigt.
Konsequenz	Prüfung der Antragsdaten auf Vollständigkeit funktioniert.

3.9. TC07 [Antrag] - Antrag anzeigen

Beschreibung	Der Antragsverwalter kann sich einen Antrag auf der Weboberfläche der Anwendung anzeigen lassen.
Vorbedingung	Es existiert mindestens ein Referat und eine Sitzung. Der Antrag ist gültig und wurde erfolgreich eingereicht.
Nachbedingung	Es existiert mindestens ein Referat und eine Sitzung. Die Daten werden korrekt abgebildet.
Benötigte Daten	Antragsdaten
Testergebnis	Ein Antragsverwalter kann einen Antrag auf der Weboberfläche anzeigen lassen und die Daten werden korrekt abgebildet.
Konsequenz	Das Anzeigen von Anträgen funktioniert.

3.10. TC08 [Antrag] - Antrag bearbeiten

Beschreibung	Der Antragsverwalter kann einen Antrag auswählen und beliebig dessen Daten verändern.
Vorbedingung	Der Antrag ist gültig und wurde erfolgreich eingereicht.
Nachbedingung	Die Daten werden korrekt abgefragt und im Nachhinein korrekt in der Datenbank aktualisiert.
Benötigte Daten	Eingereichter Antrag, Antragsdaten
Testergebnis	Ein Antragsverwalter kann einen gültigen Antrag auswählen und dessen Daten erfolgreich bearbeiten, wobei die Daten in der Datenbank korrekt aktualisiert werden.
Konsequenz	Das Bearbeiten von Anträgen funktioniert.

3.11. TC09 [Tagesordnung] - Erfolgreiches Generieren der Tagesordnung

Beschreibung	Der Systemadministrator klickt auf Sitzungsverwaltung und wählt eine Sitzung aus. Anschließend klickt er auf Tagesordnung erstellen, dabei wird ein EtherPad-Link mit Anträgen erstellt.
---------------------	---

Vorbedingung	System muss vollständig eingerichtet und funktional sein. Es muss mindestens eine Sitzung angelegt sein. Es muss mindestens ein Referat vorhanden sein. Es muss mindestens ein Antrag gestellt worden sein.
Nachbedingung	Die Tagesordnung wird erfolgreich mit allen Anträgen generiert und es öffnet sich eine EtherPad-Link.
Benötigte Daten	Referat, Sitzung, Anträge
Testergebnis	Die Tagesordnung wird erfolgreich mit allen Anträgen generiert und es öffnet sich ein EtherPad-Link.
Konsequenz	Das Generieren von Tagesordnungen und das Erstellen eines EtherPad-Dokumentes mit korrektem Inhalt funktioniert.

3.12. TC10 [Sitzung] - Sitzung vertagen

Beschreibung	Der Antragsverwalter wählt eine Sitzung aus und vertagt diese auf einen anderen Tag.
Vorbedingung	Die Sitzung muss bereits in der Datenbank hinterlegt sein.
Nachbedingung	Das Sitzungsdatum wird auf das eingegebene Datum verschoben. Anträge, die dieser Sitzung zugeordnet waren, werden mitverschoben.
Benötigte Daten	Sitzungsdaten, eingereichte Anträge
Testergebnis	Eine Sitzung kann erfolgreich auf einen anderen Tag verschoben werden, indem das Sitzungsdatum in der Datenbank aktualisiert wird.
Konsequenz	Verschieben von Sitzungen funktioniert.

3.13. TC11 [Antrag] - Antrag einer anderen Sitzung zuweisen

Beschreibung	Der Antragsverwalter vertagt einen Antrag auf eine andere Sitzung.
Vorbedingung	Der Antrag muss gültig sein und die Sitzung muss bereits in der Datenbank hinterlegt sein.
Nachbedingung	Der Antrag wird einer anderen Sitzung zugewiesen und kann in die nächste Tagesordnung übernommen werden.
Benötigte Daten	Antragsdaten, Sitzungsdaten (aktuelle & nächste Sitzung)
Testergebnis	Ein Antragsverwalter kann einen Antrag erfolgreich auf eine andere Sitzung vertagen, indem der Antrag der entsprechenden Sitzung zugewiesen wird.
Konsequenz	Das Vertagen von Sitzungen funktioniert.

3.14. TC12 [DB] - Verlust der Verbindung zur DB

Beschreibung	Die DB wird abgeschaltet und somit ein Verbindungsverlust provoziert.
Vorbedingung	System muss vollständig eingerichtet und funktional sein.
Nachbedingung	Zuvor eingereichte Anträge sind in der Datenbank abgespeichert. Es wird eine Fehlermeldung ausgegeben.
Benötigte Daten	Antragsdaten, Sitzungsdaten (aktuelle & nächste Sitzung)
Testergebnis	Zuvor eingereichte Anträge werden erfolgreich abgespeichert. Es wird keine Fehlermeldung angezeigt.
Konsequenz	Anträge werden gesichert. Es muss noch eine Fehlermeldung implementiert werden.

3.15. TC13 [Referat] - neues Referat erstellen

Beschreibung	a erstellt ein neues Referat.
Vorbedingung	System muss vollständig eingerichtet und funktional sein.
Nachbedingung	Das neu erstellte Referat wird in der Referatsübersicht angezeigt und ist in der DB abgespeichert.
Benötigte Daten	Referatsdaten (Name, E-Mail)
Testergebnis	Das neu erstellte Referat wird in der Referatsübersicht angezeigt und ist in der DB abgespeichert.
Konsequenz	Das Erstellen von Referaten funktioniert.

3.16. TC14 [Referat] - Referat Löschen

Beschreibung	a löscht Referat.
Vorbedingung	System muss vollständig eingerichtet und funktional sein. Es muss mindestens ein Referat vorhanden sein.
Nachbedingung	Das ausgewählte Referat wird nicht mehr in der Referatsübersicht angezeigt und wird aus der DB gelöscht.
Benötigte Daten	-
Testergebnis	Das ausgewählte Referat wird nicht mehr in der Referatsübersicht angezeigt und wurde aus der DB gelöscht.
Konsequenz	Das Löschen eines Referates funktioniert.

3.17. TC15 [Sitzung] - Sitzung erstellen

Beschreibung	Der a klickt auf Sitzung anlegen und wählt die zuständige Stelle und das Datum der Sitzung aus.
---------------------	---

Vorbedingung	System muss vollständig eingerichtet und funktional sein. Es muss mindestens ein Referat vorhanden sein.
Nachbedingung	Die neu erstellte Sitzung wird in der Sitzungsverwaltung angezeigt und ist in der DB vorhanden.
Benötigte Daten	Zuständige Stelle (Referat), Datum der Sitzung
Testergebnis	Die neu erstellte Sitzung wird in der Sitzungsverwaltung angezeigt und ist in der DB vorhanden.
Konsequenz	Das Erstellen von Sitzungen funktioniert.

3.18. TC16 [Sitzung] - Tagesordnung/Sitzung abschließen

Beschreibung	Der a klick auf Sitzungsverwaltung und wählt eine Sitzung aus. Anschließend klickt er auf Sitzung abschließen.
Vorbedingung	System muss vollständig eingerichtet und funktional sein. Es muss mindestens eine Sitzung angelegt sein. Es muss mindestens ein Referat vorhanden sein. Es muss mindestens ein Antrag gestellt worden sein.
Nachbedingung	Die Sitzung wird erfolgreich abgeschlossen und in der Sitzungsverwaltung als "Stattegefunden" markiert.
Benötigte Daten	Referat, Sitzung, Anträge
Testergebnis	Die Sitzung wurde erfolgreich abgeschlossen und in der Sitzungsverwaltung als "Stattegefunden" markiert.
Konsequenz	Das Abschließen der Sitzung funktioniert.

3.19. TC17 [Sitzung] - Tagesordnung/Sitzung mit nicht abgeschlossenen Anträgen abschließen

Beschreibung	Der a wählt eine Sitzung aus der Sitzungsverwaltung aus und klickt auf "Sitzung abschließen".
Vorbedingung	Das System muss vollständig eingerichtet und funktionsfähig sein. Es ist mindestens ein Antrag noch nicht behandelt.
Nachbedingung	Das Abschließen dieser Sitzung ist nicht möglich. Es wird eine Fehlermeldung angezeigt, die darauf hinweist, dass noch nicht alle Anträge behandelt wurden.
Benötigte Daten	Sitzung, Anträge
Testergebnis	Das Abschließen der Sitzung ist möglich. Es wird keine Fehlermeldung angezeigt.
Konsequenz	Die Funktionalität muss überarbeitet werden, um eine Prüfung der Anträge einzubauen.

3.20. TC18 [Sitzung] - in Zukunft liegende Sitzung abschließen

Beschreibung	Der a wählt eine Sitzung aus der Sitzungsverwaltung aus, deren Datum in der Zukunft liegt, und klickt auf "Sitzung abschließen".
Vorbedingung	Das System muss vollständig eingerichtet und funktionsfähig sein. Alle Anträge müssen behandelt worden sein.
Nachbedingung	Es wird eine Meldung angezeigt, die besagt, dass die Sitzung noch nicht abgeschlossen werden darf, da sie in der Zukunft liegt. Die Sitzung wird nicht abgeschlossen.
Benötigte Daten	Sitzung, Anträge
Testergebnis	Das Abschließen der Sitzung ist möglich. Es wird keine Fehlermeldung angezeigt.
Konsequenz	Die Funktionalität muss überarbeitet werden, um eine Prüfung des Datums einzufügen.

Kapitel 4. Konsequenzen

Die identifizierten Fehler und Abweichungen müssen behoben werden, um die Software zu verbessern und sicherzustellen, dass sie ordnungsgemäß funktioniert. Es sollte eine erneute Durchführung der fehlgeschlagenen Testfälle erfolgen, um sicherzustellen, dass die Fehler behoben wurden und die gewünschten Ergebnisse erzielt werden. Es sollte auch eine Überprüfung der Testabdeckung durchgeführt werden, um sicherzustellen, dass alle relevanten Testfälle abgedeckt sind und dass keine weiteren Fehler vorhanden sind. Bei der Übergabe der Software sollten Abnahmetests durchgeführt werden, um sicherzustellen, dass die Software den Anforderungen des Kunden entspricht und die erwarteten Ergebnisse liefert.