

# Testdokumentation: Antragsverwaltungstool (I3)

## Inhaltsverzeichnis

Testplan .....	1
Teststrategie .....	1
Testumgebung .....	2
Testdokumentation .....	2
Testorganisation .....	2
Testausführungsplan .....	2
Testkonzept .....	2
Testfallbeschreibung .....	3
Testergebnisse .....	4
Allgemeine Informationen .....	4
Fehlerliste: .....	4
Verbesserungsvorschläge: .....	4
Notizen: .....	5
Rechtschreibung/Grammatik: .....	5

## Testplan

### Teststrategie

Eines der höchsten Risiken von Fehlfunktionen der Software ist der Verlust von Antragsdaten. Wenn Anträge gar nicht oder falsch abgespeichert werden, dann kann dies für große Probleme beim StuRa sorgen und macht die Software unzuverlässig und damit auch unbrauchbar.

Deshalb ist es essenziell, dass das Einreichen und Bearbeiten von Anträgen ohne Datenverlust oder ungewollte Datenmanipulation funktioniert. Ebenfalls dürfen bei der Antragsbearbeitung keine Daten falsch angezeigt werden, da dies für Fehlinformationen und -entscheidungen sorgen würde.

Problematisch kann auch ein Angriff einer böswilligen Person sein. Da wir keine Skripte und log-Dateien gegen DOS-Angriffe und Antragspppamming entwickelt besteht an dieser Stelle das Risiko, dass die Arbeit des StuRa z.B. durch einen Serverabsturz behindert wird. Ebenfalls könnten SQL-Injections möglich sein, welche dafür sorgen, dass Daten aus der Datenbank gelöscht werden. Da es bisher keine Maßnahmen gegen solche Angriffe gibt können auch keine solchen Sicherheitsmaßnahmen getestet werden.

# Testumgebung

Zum Testen wird keine besondere Testumgebung benötigt. Da das System auf dem HTW-Server gehostet wird braucht man lediglich eine Internetverbindung, um das Tool über den Browser aufzurufen und zu testen/nutzen.

Bevor das Tool auf dem Server gehostet wurde war es notwendig die Systemvoraussetzungen (siehe Betriebsdokumentation) lokal auf dem eigenen Gerät zu schaffen. Dies hat den Test- und Entwicklungsprozess erheblich erschwert bzw. unmöglich gemacht.

Die Tests werden ausschließlich manuell durchgeführt. Mit Skripten o.Ä. automatisierte Test werden aktuell nicht benötigt.

## Testdokumentation

Die Testergebnisse werden in einer Tabelle (siehe test\_log) protokolliert. Es werden der Name des Tests, das Datum an dem er durchgeführt wurde, das Ergebnis und sonstige Anmerkungen/Notizen in der Tabelle notiert. Sonstige Fehler und Verbesserungsvorschläge werden in der Fehlerliste dokumentiert.

## Testorganisation

Die Test Teammitglieder prüfen alle wichtigen use-cases auf Fehlfunktionen, optimalen Workflow für den Nutzer und sonstige Ungenauigkeiten (wie z.B. Rechtschreibfehler). Nachdem die Tests durchgeführt sind, werden alle Funde möglichst zeitnah mit den für die Implementierung zuständigen Teammitgliedern besprochen. Diese setzen dann die notwendigen Änderungen um.

Während der Implementierung werden zudem schon viele Funktionen parallel von den Entwicklern getestet. Dies sorgt dafür, dass kaum große Probleme bei den Tests gefunden werden bzw. das ein Großteil der Fehlfunktionen bereits bekannt ist.

## Testausführungsplan

Alle vollständig implementierten use-cases werden von den Testern ausführlich getestet, sobald das Tool auf dem Server installiert ist und bereit steht.

## Testkonzept

Die wichtigsten zu testenden Komponenten waren die Weboberfläche (Front-End), die Datenbank und diverse Skripte zur Datenübermittlung und -verarbeitung.

Es wurden alle essenziellen Funktionen des Tools ausführlich getestet. Dazu gehört das Einreichen von Anträgen, die Suche und Anzeige von bereits eingereichten Anträgen und deren Bearbeitung.

Zum testen des use-cases "Antrag einreichen" wurde mittels der Antragsmaske verschiedene Testanträge in allen Antragskategorie eingereicht. Dabei wurde versucht diese mit fehlenden Daten einzureichen, um die Checks für leere Eingabefelder (z.B. Antragstitel, Name und E-Mail) zu testen.

Anschließend wurde über den Reiter "intern" das Suchen, das Anzeigen und die Bearbeitung von bereits eingereichten Anträgen getestet. Dazu wurde lediglich bei der Antragssuche eine Stelle ausgewählt und die Suche abgesendet.

Zum Testen der Anzeige und Bearbeitung von Anträgen wurden für verschiedene Anträge mittels der Antragsnummer (sowohl bereits vorhandene als auch nicht existierende) die Daten angefordert. Diese wurden dann mit den abgesendeten Daten abgeglichen. Zum Testen der Bearbeitungsfunktion wurden alle Daten eines Antrags bearbeitet und die Änderungen abgesendet. Danach wurden wieder die übermittelten Daten abgeglichen.

Ein Teil der Funktionstests ist bereits während der Implementierung durchgeführt worden. Ausführliche und tiefgehende Tests auf Fehlfunktionen und Verbesserungsmöglichkeiten sind dann durch das Team Test getätigt worden.

## Testfallbeschreibung

Testobjekt	Testfall	Vorbedingung	Daten	erwartetes Ergebnis
Weboberfläche , Datenbank, Django Skripte	Valide Logindaten akzeptiert	Nutzerdaten wurden angelegt, aktuell nicht eingeloggt	Nutzerdaten	Nutzer ist angemeldet und hat Zugriff auf "Intern"
Weboberfläche , Datenbank, Django Skripte	Falsche Logindaten abgelehnt	aktuell nicht eingeloggt	falsche Nutzerdaten	Login schlägt fehl
Weboberfläche , Datenbank, Skripte	Vollständig ausgefüllter Antrag eingereicht	alle Pflichtfelder sind ausgefüllt	Antragsdaten	Antrag wird an Datenbank übermittelt
Weboberfläche , Skripte	Unvollständige r Antrag abgelehnt	min. ein Pflichtfeld nicht ausgefüllt	unvollständige Antragsdaten	Antrag wird nicht eingereicht. Nutzer wird auf nicht ausgefüllte Pflichtfelder hingewiesen
Weboberfläche , Datenbank, Skripte	Antrag wird korrekt angezeigt	Daten für existierenden Antrag werden angefordert	Antragsdaten	Alle Daten des Antrags werden korrekt und vollständig in Bearbeitungsmaske angezeigt
Weboberfläche , Django Skripte	externer Nutzer kann keine Anträge einsehen	aktuell nicht eingeloggt	-	Link "intern" wird nicht in Navigationsleiste angezeigt

Testobjekt	Testfall	Vorbedingung	Daten	erwartetes Ergebnis
Weboberfläche , Datenbank, Skripte	Anträge lassen sich bearbeiten	Antrag existiert	Antragsnumme r, Antragsdaten	Daten können bearbeitet werden. Daten werden erfolgreich an Datenbank übermittelt.

# Testergebnisse

## Allgemeine Informationen

- alle Test wurden manuell durchgeführt
- Testdaten für Anträge (meist) nur zufällige Tastatureingaben

Testfall	Datum	Ergebnis	Notizen
valide Login-Daten akzeptiert	28.06.2021	bestanden	-
falsche Login-Daten abgelehnt	28.06.2021	bestanden	-
Vollständig ausgefüllter Antrag eingereicht	28.06.2021	bestanden	-
unvollständiger ausgefüllter Antrag abgelehnt	28.06.2021	nicht bestanden	Anträge lassen sich mit unausgefüllten Pflichtfeldern (z.B. zuständige Stelle, Kostenposition) einreichen
Antrag wird korrekt angezeigt	28.06.2021	Warnung	alle Antragsdaten außer Stelle
externen Nutzer kann keine Anträge einsehen	28.06.2021	bestanden	nur über Reiter "intern" möglich → Login wird benötigt
Anträge lassen sich bearbeiten	28.06.2021	Warnung	alle Daten lassen sich bearbeiten, aber Änderung an "absoluter Mehrheit" wird nicht gespeichert
Antrag wird korrekt angezeigt	01.07.2021	bestanden	Stelle wird nun auch korrekt ausgegeben

## Fehlerliste:

- Daten für ungültige Antragsnummer anfordern → Django unexpected error 404
- einige Felder bei Antragsbearbeitung ausgegraut

## Verbesserungsvorschläge:

- Antragsbearbeitung etwas versteckt: Bearbeitungsfunktion mehr hervorheben → leichter verständlich für Anwender
- feedback wenn Antrag abgeschickt wurde?

- Antrag mit unverändertem Antragstext abschickbar?

## Notizen:

- Name kein Leerzeichenfilter
- Antragsbearbeitung
  - Leerzeichen vor Antragsnummer, unterhalb von Daten anfordern
  - zuständige Stelle "leer" (besser aktuelle Stelle bereits ausgewählt?)
  - abgeschlossene Anträge können weiterhin bearbeitet werden?
  - Textboxen in Größe veränderbar: gewollt?
- Antragsnummer ohne korrektes Datum
- Anträge löschen?
- Kostenposition im Haushaltsplan: großes Textfeld notwendig?
- Antrag ohne Kostenposition möglich?

## Rechtschreibung/Grammatik:

- StuRa Antragsverwaltung: "notwendig,[ ]wenn", "konsultiert werden[.]"
- besondere MeHrheit
- Beschlusstext ":",