

SE I - Belegabgabe

Antragsverwaltungstool (I3)

2021-02-11 15:29:30 +0100

Richard Frieze <richard.frieze@htw-dresden.de>; Andreas Gröber <andreas.groeber@htw-dresden.de>; Jasmin Henn <jasmin.henn@htw-dresden.de>; Ruben-David Kraus <s80350@htw-dresden.de>; Richard Müller <richard.mueller2@htw.dresden.de>; Duy Tien Nguyen <s80287@htw-dresden.de>; Amanda Wer Estrada <amanda.werestrada@htw-dresden.de>; Jacob Flaskamp <s81424@htw-dresden.de>]

Inhaltsverzeichnis

Technische Spezifikation	1
1. Vision: Antragsverwaltungstool (I3)	2
1.1. Einführung	2
1.2. Positionierung	2
1.3. Stakeholder Beschreibungen	3
1.4. Produkt-/Lösungsüberblick	4
1.5. Zusätzliche Produktanforderungen	5
2. Use-Case Model: Antragsverwaltungstool (I3)	7
2.1. Allgemeine Informationen	7
2.2. Identifizierte Use-Cases	7
2.3. Use-Case 01: Wahlantrag stellen	7
2.4. Use-Case 07: Tagesordnung ansehen	14
2.5. Use-Case 08: Tagesordnung bearbeiten	16
2.6. Use-Case 09: Antrag bearbeiten	17
2.7. Use-Case 10: Antrag ablehnen	18
2.8. Use-Case 11: Antrag vertagen	19
2.9. Use-Case 12: Antrag ablehnen	21
2.10. Use-Case 13: Antrag vertagen	22
3. System-Wide Requirements: {Antragsverwaltungstool (I3)}	25
3.1. Einführung	25
3.2. Systemweite funktionale Anforderungen	25
3.3. Qualitätsanforderungen für das Gesamtsystem	26
3.4. Zusätzliche Anforderungen	27
4. Glossar: Antragsverwaltungstool (I3)	28
4.1. Einführung	28
4.2. Begriffe	28
4.3. Abkürzungen und Akronyme	30
4.4. Verzeichnis der Datenstrukturen	30
5. Domain Model: Antragsverwaltungstool (I3)	31
5.1. Allgemeine Informationen	31
Projektdokumentation	32
6. Projekt Plan: Antragsverwaltungstool (I3)	33
6.1. Einleitung	33
6.2. Projektorganisation	33
6.3. Projektpraktiken and Messwerte	34
6.4. Projektmeilensteine und -ziele	34
6.5. Bereitstellung der Software	34
6.6. Gewonnene Erkenntnisse	34

6.7. Attribute	35
6.8. Risiken	35
7. Iteration Plan 01: I3_Antragsverwaltungstool	39
7.1. Meilensteine	39
7.2. Wesentliche Ziele	39
7.3. Aufgabenzuordnung	39
7.4. Probleme (optional)	40
7.5. Bewertungskriterien	40
7.6. Assessment	40
8. Iteration Plan 02: I3_Antragsverwaltungstool	41
8.1. Meilensteine	41
8.2. Wesentliche Ziele	41
8.3. Aufgabenzuordnung	41
8.4. Bewertungskriterien	42
8.5. Assessment	42
9. Iteration Plan 03: I3_Antragsverwaltungstool	43
9.1. Meilensteine	43
9.2. Wesentliche Ziele	43
9.3. Aufgabenzuordnung	43
9.4. Bewertungskriterien	43
9.5. Assessment	44
Liste der Anfallenden Aufgaben	45
Entwurfsdokumentation	48
Architecture Notebook: Antragsverwaltungstool (I3)	49
10. Zielsetzung	50
11. Entwurfsziel und Philosophie	51
12. Annahmen und Abhängigkeiten	52
13. Architektonisch bedeutende Anforderungen (ergibt sich aus Use-Cases und Furbs+)	53
14. Entscheidungen, Einschränkungen und Begründungen (Programmiersprachen „Bibliotheken, Framworks, DBS-Managementsystem usw + Begründung)	54
15. Architekturmechanismen (Technische Umsetzung die sich aus Anforderung ergibt, aus Anforderungen welche Technik ich brauche)	55
16. Abstraktion	56
17. Architektonischer Rahmen	57
18. Architektonische Ansichten	58
19. Test Cases: Antragsverwaltungstool (I3)	59
19.1. Allgemeine Informationen	59
19.2. Test Cases	59

Technische Spezifikation

- Vision
- Use Case Model (inkl. Wireframes, sofern vorhanden)
- System-wide Requirements
- Glossar
- Domänenmodel

1. Vision: Antragsverwaltungstool (I3)

Richard Friese <richard.friese@htw-dresden.de>; Andreas Gröber <andreas.groeber@htw-dresden.de>; Jasmin Henn <jasmin.henn@htw-dresden.de>; Ruben-David Kraus <s80350@htw-dresden.de>; Richard Müller <richard.mueller2@htw-dresden.de>; Duy Tien Nguyen <s80287@htw-dresden.de>; Amanda Wer Estrada <amanda.werestrada@htw-dresden.de>; Jacob Flaskamp <s81424@htw-dresden.de>

1.1. Einführung

Der Zweck dieses Dokuments ist es, die wesentlichen Bedarfe und Funktionalitäten des I3 Antragsverwaltungstool zu sammeln, zu analysieren und zu definieren. Der Fokus liegt auf den Fähigkeiten, die von Stakeholdern und adressierten Nutzern benötigt werden, und der Begründung dieser Bedarfe. Die Details, wie das I3 Antragsverwaltungstool diese Bedarfe erfüllt, werden in der Use-Case und Supplementary Specification beschrieben.

1.1.1. Zweck

Der Zweck dieses Dokuments ist es, die wesentlichen Anforderungen an das System aus Sicht und mit den Begriffen der künftigen Anwender zu beschreiben.

1.1.2. Gültigkeitsbereich (Scope)

Dieses Visions-Dokument bezieht sich auf das I3 Antragsverwaltungstool, das von Team I3 entwickelt wird. Das System wird es dem Stura der HTW Dresden erlauben, Anträge zu verwalten, indem Antragsformulare vervollständigt werden und automatisch in die Tagesordnung der nächsten Sitzung eingeordnet werden, um damit einen schnelleren und leichteren Antragsverwaltungsprozess zu ermöglichen. Diese browserbasierte Anwendung soll das Problem der Unvollständigkeit der Formulare lösen und dem Stura informieren, wenn Anträge erstellt werden.

1.1.3. Definitionen, Akronyme und Abkürzungen

siehe Glossar [glossary.pdf](#)

1.1.4. Referenzen

(hier externe Verweise zu anderen Dokumenten, Quellen, Standards etc. einfügen, sofern notwendig)

1.2. Positionierung

1.2.1. Fachliche Motivation

Der Stura der HTW Dresden verwaltet Antragsformulare noch analog. Wenn Bedürfnisse in den einzelnen Referaten entstehen, müssen diese einen Antrag stellen. Das Präsidium wird benachrichtigt und der Antrag bekommt eine Identifizierungsnummer. Dann werden Anträge

manuell in die Tagesordnung nach einer bestimmten Priorität eingeordnet, um verarbeitet zu werden. Wenn der Antrag nicht in der Sitzung bearbeitet wird, wird er vertagt. Aufgrund dieses Prozesses verliert der Stura den Überblick und manche Anträge werden nicht richtig ausgefüllt. Die unbearbeiteten Anliegen gehen durcheinander und der Stura wird nicht benachrichtigt, wenn ein neues Formular erstellt wird. Die angestrebte Verbesserung hat als Ziel die Einführung eines browserbasierten Tools, um das Erstellen und Verwalten von Anträgen übersichtlich, einfach und automatisch zu gestalten.

1.2.2. Problem Statement

Das Problem	die unübersichtliche Bearbeitung der Anträge
betrifft	alle Mitglieder des Sturas, das Referatskollegium, das Präsidium sowie Sprecher/innen.
Die Auswirkung davon ist	die suboptimale Leistung des Sturas und das langsame Bearbeiten von Anträgen
eine erfolgreiche Lösung wäre	eine Anwendung um korrekt ausgefüllte und nummerierte Formulare zu erstellen und gemäß Priorisierung in der Tagesordnung der folgenden Sitzung einzuordnen. Dies entspricht einer extrem beschleunigten, transparenten und vereinfachten Überarbeitung von Anträgen. Das Präsidium wird entlastet.

1.2.3. Positionierung des Produkts

Für	den Stura der HTW Dresden
der	Anträge für Wahlen und Beschlüsse verwaltet
Die Lösung ist eine	browserbasierte Anwendung
Das	digitales Ausfüllen von Antragsformularen
Im Gegensatz zur	manuelle Verwaltung der Anträge
Unser Produkt	erstellt Antragsvorlagen teilautomatisch und verknüpft die ausgefüllten Anträgen mit der Tagesordnung für folgende Sitzungen.

1.3. Stakeholder Beschreibungen

1.3.1. Zusammenfassung der Stakeholder (Mitglieder des Sturas)

Name	Beschreibung	Verantwortlichkeiten
Referatskollegium	beratende gewählte Mitglieder des Sturas, Studenten	Nutzer der Anwendung und stellen Anträge auf Beschlüsse und Personenwahlen

Name	Beschreibung	Verantwortlichkeiten
Sprecher/innen	beratende gewählte Mitglieder des Sturas, Studenten	Nutzer der Anwendung und stellen Anträge für Beschlüsse und Personenwahlen, können selber Beschlüsse fassen
Präsidium	die Leiter des Sturas, Studenten	sie haben administrative Rechte und sind gleichzeitig Nutzer der Anwendung, Verwaltung der Anträge
Gesetzgeber	gibt rechtliche Rahmenbedingungen vor (Datenschutz)	überwacht die Einhaltung des Gesetzes
jede studentische Person der HTW Dresden (potentielle Antragssteller)	Studenten	jeder mit Interesse an Amt und/oder Mithilfe im Stura

1.3.2. Benutzerumgebung

1. Antragssteller

- Anzahl der Studenten variiert
- bei wiederholter Nutzung des Tools verringert sich die Bearbeitungszeit, da der Nutzer sich damit auskennt
- Student ist an einem Antragprozess interessiert, da er intuitiv und aufwandsarm ist
- die Anwendung kann mit jedem Browser durchgeführt werden

2. Antragsverwalter

- feste, berechnigte Mitglieder des Sturas
- Arbeitsaufwand hängt vom Formular ab (Richtigkeit und Vollständigkeit)
- Darstellung auf einer Desktopansicht

1.4. Produkt-/Lösungsüberblick

1.4.1. Bedarfe und Hauptfunktionen

Bedarf	Priorität	Features	Geplantes Release
Vorlagen der Anträge digitalisieren	sehr hoch	Drop-down Menü, Ankreuzfunktion, automatische Gererierung der Kennziffer des Antrags	SS2021

Bedarf	Priorität	Features	Geplantes Release
Wiedervorlage von vertagten Anträgen	hoch	Verknüpfung des Tools mit der Tagesordnung während der Versammlung, Zugang zur Liste von vertagten Anträgen	SS2021
Automatische Generierung der Tagesordnung	hoch	Verknüpfung mit der Anwendung, Wahlen werden priorisiert	SS2021
Verlinkung zum Haushaltsplan	niedrig	Weiterleitung zum Haushaltsplan des Sturas	

1.5. Zusätzliche Produktanforderungen

- Die Anwendung soll eine ähnliche Interface zu Antragsgrün haben.
- Kein Java
- Die browserbasierte Anwendung muss auf Linux und Unix laufen.
- freie Lizenz benutzen (zum Beispiel GPL, MIT, Apache).
- gern auch in einem bekannten Webframework
- Robustheit eher wichtig bei erstellen neuer Anträge
- Ausfalltoleranz nicht abschätzbar
- Benutzerfreundlichkeit hoch anzusetzen da eher unerfahrene Nutzer
- Mobile Variante geringe Priorität
- Eingabeüberprüfung sehr wichtig
- Information/Ergebnis per E-Mail an Wahlkandidaten

Anforderung	Priorität	Geplantes Release
Vorlagen der Anträge digitalisieren	sehr hoch	SS2021
Verknüpfung der Anwendung mit der Tagesordnung	hoch	SS2021
Priorisierung von Anträge	hoch	SS2021
Benachrichtigung von neue Mitglieder	niedrig/extra	SS2021
Kein Java	hoch	-
Browserbasierte Anwendung	sehr hoch	SS2021
Robustheit	hoch	SS2021
Ausfalltoleranz	nicht abschätzbar	-
Mobile Variante	niedrig	SS0201

Anforderung	Priorität	Geplantes Release
Eingabeüberprüfung	hoch	SS2021
Information per E-Mail an Wahlkandidaten	niedrig	SS02021
Hilfe-Funktion direkt auf Website	niedrig	SS20201

2. Use-Case Model: Antragsverwaltungstool (I3)

Richard Frieze <richard.frieze@htw-dresden.de>; Andreas Gröber <andreas.groeber@htw-dresden.de>; Jasmin Henn <jasmin.henn@htw-dresden.de>; Ruben-David Kraus <s80350@htw-dresden.de>; Richard Müller <richard.mueller2@htw-dresden.de>; Duy Tien Nguyen <s80287@htw-dresden.de>; Amanda Wer Estrada <amanda.werestrada@htw-dresden.de>; Jacob Flaskamp <s81424@htw-dresden.de>

2.1. Allgemeine Informationen

2.2. Identifizierte Use-Cases

Für Mitglieder

- UC01: Wahlantrag stellen
- UC02: Antrag für Ressourcen stellen
- UC03: Offene Anträge ansehen
- UC04: Änderungsantrag vom Antragssteller
- UC05: Tagesordnung anzeigen
- Add-on: (Haushaltsplan ansehen)
- Add-on: (Email an gewählten Kandidat)

Fürs Präsidium

- UC06: Offene Anträge ansehen
- UC07: Tagesordnung ansehen
- UC08: Tagesordnung bearbeiten
- UC09: Antrag bearbeiten
- UC10: Antrag ablehnen
- UC11: Antrag vertagen
- Add-on: (Haushaltsplan ansehen)
- Add-on: (Information Mitgliederverwaltung über neues Mitglied)

2.3. Use-Case 01: Wahlantrag stellen

2.3.1. Kurzbeschreibung

Der Use Case beschreibt den kompletten Ablauf der Wahlantragsbearbeitung.

2.3.2. Kurzbeschreibung der Akteure

externer Antragssteller:

Möchte auf einen Posten im Stura kandidieren.

interner Antragssteller:

Möchte auf einen weiteren/anderen Posten im Stura kandidieren.

2.3.3. Vorbedingungen

Antragsteller ist beliebige natürliche Person und hat das Antragsverwaltungstool für seine Antragstellung gewählt.

2.3.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Antragssteller das Antragsverwaltungstool geöffnet hat.
2. WHILE Antrag ist nicht vollständig
 1. Antragssteller gibt dem Antrag einen Namen
 2. Antragsteller gibt Datum ein
 3. System generiert Antragskennziffer
 4. Antragsteller wählt seine Rolle aus
 5. Antragsteller beantwortet Fragen
3. Antragsteller löst Antragsbearbeitung aus
4. System erstellt TOP
5. Der Use Case ist abgeschlossen

2.3.5. Alternative Abläufe

<Alternativer Ablauf 1>

(a), (c) und (d) Antworten entsprechen nicht den Gültigkeitsregeln

1. System fordert Antragsteller auf, diese zu berichtigen
2. Antragsteller berichtigt Angaben, Fortsetzung Standardablauf

2.3.6. Wesentliche Szenarios

Erfolgreiche Standardbeantragung

Sc01: Antragssteller X füllt den Antrag auf Personenwahl aus. Antrag erfolgreich abgegeben und weitergeleitet sowie in Tagesordnung vermerkt.

Antragssteller wählt nicht gültige Eingabe bzw. füllt nicht alle Pflichtfelder aus

Sc02: Antragssteller wählt keine gültige Eingabe oder vergisst ein Feld auszufüllen. System fordert ihn auf die Eintragung zu ändern oder die fehlende zu ergänzen.

2.3.7. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

1. Antragssteller hat Formular komplett ausgefüllt
2. Alle eingegebenen Daten und ausgewählten Optionen gespeichert
3. Antrag wurde an Präsidium weitergeleitet
4. Antrag bekommt TOP und wird in TO aufgenommen

2.3.8. Besondere Anforderungen

System-wide Requirements == Use-Case 02: Antrag für Ressourcen stellen === Kurzbeschreibung
Der Use Case beschreibt den kompletten Ablauf der Beantragung eines Antrages für Ressourcen (Sachantrag).

2.3.9. Kurzbeschreibung der Akteure

Antragssteller:

Möchte Antrag stellen, sodass der Bedarf an Ressourcen in den einzelnen Referaten vom Stura gedeckt wird.

Vorstand der Geschäftsführung:

Stellt Antrag für Beschlüsse, können selber Beschlüsse fassen.

Referatskollegium:

Stellt Anträge auf Beschlüsse.

2.3.10. Vorbedingungen

1. Keine Anmeldung benötigt
2. Antragsteller ist Mitglied des Sturas und/oder eines weiteren Referats.

2.3.11. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Antragssteller das Antragsverwaltungstool geöffnet hat.
2. Akteur wählt geforderte Anträge aus. (Vorlage des Sachantrags auf der Website des Sturas verfügbar)
3. WHILE Antrag ist nicht vollständig

- a. Antragssteller gibt dem Antrag einen Namen
 - b. System generiert Antragskennziffer
 - c. Antragsteller gibt seine Rolle ein
 - d. Antragsteller beschreibt das Ressourcenbedürfnis
4. Antragssteller füllt den Rest des Formulars aus
 5. Antragsteller löst Antragsbearbeitung aus
 6. Antragsteller reicht Antrag ein
 7. System erstellt TOP
 8. Der Use Case ist abgeschlossen

2.3.12. Alternative Abläufe

Alternativer Ablauf 1

Wenn die Antworten des Antragsstellers im Schritt (a),(c) und (d) des Standardablaufs nicht den Gültigkeitsregeln entsprechen, dann:

1. System fordert Antragsteller auf, diese zu berichtigen
2. Antragsteller berichtigt Angaben, Fortsetzung Standardablaufs

2.3.13. Wesentliche Szenarios

Sc01: Erfolgreiche Beantragung

1. Antragssteller füllt den Antrag für Ressourcen aus.
2. Antrag erfolgreich abgegeben und weitergeleitet.

Sc02: Antragsteller gibt nicht gültige Antworten ein bzw. füllt nicht alle Pflichtfelder aus

1. Antragsteller wählt keine gültige Eingabe oder vergisst ein Feld auszufüllen.
2. System fordert ihn auf die Eintragung zu ändern oder die fehlende zu ergänzen.

2.3.14. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

- Antragsteller hat Formular komplett ausgefüllt.
- Alle eingegebenen Daten und ausgewählten Optionen gespeichert.
- Antrag wurde an Präsidium weitergeleitet.
- Antrag bekommt TOP und wird in TO aufgenommen.

2.3.15. Besondere Anforderungen

[System-wide Requirements](#) == Use-Case 03: Offene Anträge ansehen == Kurzbeschreibung Der Use

Case beschreibt das Vorgehen zum Ansehen von offenen Anträgen.

2.3.16. Kurzbeschreibung der Akteure

Antragssteller:

Der Antragssteller möchte wissen, ob sein Antrag bearbeitet wurde.

Präsidium:

Das Präsidium möchte wissen, welche Anträge noch zu bearbeiten sind.

2.3.17. Vorbedingungen

Antragsteller hat einen Antrag eingereicht. Es wurde schonmal ein Antrag eingereicht.

2.3.18. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn Akteur den Antrag aufrufen möchte.
2. Akteur ruft Antragsformulartool auf.
3. Akteur wählt geforderten Antrag aus.

2.3.19. Wesentliche Szenarios

Sc01: Erfolgreiche Einsicht

Akteur kann den gewünschten Antrag aufrufen.

Sc02: Gescheiterte Einsicht

Es ist kein Antrag vorhanden. Akteur kann sein Bedürfnis nicht nachgehen.

2.3.20. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

- Anträge müssen unverändert sein

2.3.21. Besondere Anforderungen

System-wide Requirements == Use-Case 04: Änderungsantrag vom Antragsteller ==
Kurzbeschreibung Der Use Case beschreibt das Vorgehen zum Bearbeiten von Anträgen durch den Antragsteller.

2.3.22. Kurzbeschreibung der Akteure

Antragssteller:

Der Antragsteller möchte einen Änderungsantrag stellen.

2.3.23. Vorbedingungen

Antragsteller hat bereits einen Antrag eingereicht.

2.3.24. Standardablauf (Basic Flow)

1. Antragsteller öffnet Antrag im Antragsverwaltungstool
2. Antragsteller füllt Änderungsantrag aus.
3. Antragsteller gibt Änderungsantrag ab.

2.3.25. Wesentliche Szenarios

2.3.26. Änderungsantrag eingereicht

Sc01: Der Antragssteller hat Änderungsantrag vollständig ausgefüllt.

2.3.27. Änderungsantrag uneindeutig

Sc02: Die Angaben des Antragsstellers sind uneindeutig. Daraufhin setzt sich das Präsidium mit Ihm in Verbindung.

2.3.28. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

- Antragssteller hat Formular komplett ausgefüllt
- Alle eingegebenen Daten und ausgewählten Optionen gespeichert.
- Antrag wurde an Präsidium weitergeleitet.
- Antrag bekommt TOP und wird in TO aufgenommen.

2.3.29. Besondere Anforderungen

[System-wide Requirements](#) == Use-Case 05: Tagesordnung anzeigen === Kurzbeschreibung Der Use Case beschreibt das Vorgehen zum Ansehen der aktuellen Tagesordnung für die kommende Sitzung.

2.3.30. Kurzbeschreibung der Akteure

Antragssteller:

Der Antragsteller möchte wissen ob sein Antrag in der aktuellen TO ist und sich generell informieren.

Präsidium:

Das Präsidium möchte sich auf nächste Sitzung vorbereiten oder sich generell informieren.

alle weiteren Interessenten an der Studentischen Vertretung:

Alle weiteren anderen Interessenten an der Studentischen Vertretung möchten sich informieren.

2.3.31. Vorbedingungen

Es ist eine Sitzung des Stura geplant und es wurden Anträge eingereicht sowie TOPs festgelegt.

2.3.32. Standardablauf (Basic Flow)

1. Antragsverwaltungstool aufrufen
2. Akteur sieht sich TO an

2.3.33. Wesentliche Szenarios

TO aufrufen

Sc01: Akteur ruft im Antragsverwaltungstool die Option Tagesordnung auf.

TO nicht vorhanden

Sc02: Akteur möchte die TO aufrufen, jedoch ist die Option nicht möglich da keine Sitzung mit Tagesordnungspunkten ansteht.

2.3.34. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

- Neue TO muss abgespeichert und online gestellt werden.

2.3.35. Besondere Anforderungen

[System-wide Requirements](#) == Use-Case 06: Offene Anträge ansehen === Kurzbeschreibung Der Use Case beschreibt das Vorgehen zum Ansehen von offenen Anträgen im internen Bereich.

2.3.36. Kurzbeschreibung der Akteure

Antragsverwalter

Der Antragsverwalter möchte wissen, ob es Anträge zu bearbeiten gibt.

Präsidium

Will die TO im Voraus sehen und die Anträge, die schon verknüpft wurden.

Vorstand der Geschäftsführung

Will die TO im Voraus sehen und die Anträge, die schon verknüpft wurden. Besonders für abgesnete TO.

2.3.37. Vorbedingungen

1. Antragsverwalter ist angemeldet (hat Zugang, also er hat administrative Rechte und ist gleichzeitig Nutzer der Anwendung.)
2. Es wurden schon Anträge eingereicht.

2.3.38. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn Akteur den Antrag aufrufen möchte.
2. Akteur ruft Antragsformulartool auf.
3. Akteur ruft Liste der eingereichten Anträge auf.
4. Der Use Case ist abgeschlossen.

2.3.39. Wesentliche Szenarios

Erfolgreiche Einsicht

Interessierter kann den gewünschten Antrag aufrufen und ansehen.

Gescheiterte Einsicht

Es ist kein Antrag vorhanden. Akteur kann sein Bedürfnis nicht nachgehen.

2.3.40. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

- Anträge unverändert sein
- Antragsnummer ist korrekt
- Antrag hat einen Status: zum Präsidium eingereicht/in Bearbeitung/ freigegeben

2.3.41. Besondere Anforderungen

[System-wide Requirements](#)

2.4. Use-Case 07: Tagesordnung ansehen

2.4.1. Kurzbeschreibung

Der Use Case beschreibt das Vorgehen zum Ansehen der aktuellen Tagesordnung für die kommende Sitzung im internen Bereich.

2.4.2. Kurzbeschreibung der Akteure

Antragsverwalter:

Will die TO im Voraus sehen und die Anträge, die schon verknüpft wurden.

Präsidium:

Will die TO im Voraus sehen und die Anträge, die schon verknüpft wurden.

Vorstand der Geschäftsführung:

Will die TO im Voraus sehen und die Anträge, die schon verknüpft wurden.

2.4.3. Vorbedingungen

1. Benutzer ist angemeldet.
2. Es ist eine Sitzung des STURA geplant.
3. Es wurden Anträge eingereicht.
4. Anträge wurden schon mit TO verknüpft sowie TOPs festgelegt.
5. TO hat 3 mögliche Stadien:
 - a. automatisch generiertes internes Entwurf
 - b. dynamisch live Version der TO(verändert und aktualisiert sich fortlaufend) intern veröffentlicht
 - c. generierte TO extern veröffentlicht nach Absendung der Einladung zur nächsten Sitzung (nur mit Anträge die bereits vom Präsidium abgesegnet wurden und konform sind). TO bereit zur Besprechung.

2.4.4. Standardablauf (Basic Flow)

1. Antragsverwaltungstool aufrufen
2. Akteur sieht sich TO an.
3. Der Use Case ist abgeschlossen.

2.4.5. Wesentliche Szenarios**TO aufrufen**

Der Akteur ruft im Antragsverwaltungstool die Option Tagesordnung auf.

TO nicht vorhanden

Der Akteur möchte die TO aufrufen, jedoch ist die Option nicht möglich da keine Sitzung mit TOPs ansteht.

2.4.6. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

- Existenz einer TO

2.4.7. Besondere Anforderungen

System-wide Requirements

2.5. Use-Case 08: Tagesordnung bearbeiten

2.5.1. Kurzbeschreibung

Präsidium möchte die Tagesordnung bearbeiten.

2.5.2. Kurzbeschreibung der Akteure

Präsidium:

Ist nicht einverstanden mit der aktuellen TO.

2.5.3. Vorbedingungen

Es muss eine Tagesordnung geben und diese entspricht aktuell nicht den Bedürfnissen der aktuellen Sitzung.

2.5.4. Standardablauf (Basic Flow)

1. Aufruf des Antragsverwaltungstool durch das Präsidium.
2. Präsidium bearbeitet TO
3. TO abspeichern
4. TO online stellen

2.5.5. Wesentliche Szenarios

Sc01: TO vertagen Präsidium möchte gesamte TO vertagen. Dies geschieht über das Vertagen aller TOP.

Sc02: TOP löschen Präsidium möchte n TOPs löschen. Präsidium wählt zu löschende TOPs aus und entfernt diese.

Sc03: TOP vertagen Präsidium möchte n TOPs vertagen. Präsidium wählt die zu vertagenden TOPs und verschiebt diese auf eine andere TO.

Sc04: TOP verschieben Präsidium möchte die Reihenfolge der TO verschieben. Präsidium wählt dafür zu verschiebende TOP aus und ändert die Reihenfolge.

Sc05: TOP neu gewichten Präsidium möchte Priorität des TOP ändern. Präsidium wählt für TOP neue Prioritätszahl.

2.5.6. Nachbedingungen

- Neue TO muss abgespeichert und online gestellt werden.

2.5.7. Besondere Anforderungen

System-wide Requirements

2.6. Use-Case 09: Antrag bearbeiten

2.6.1. Kurzbeschreibung

Der Use Case beschreibt das Vorgehen zum Bearbeiten von Anträgen im internen Bereich, sowohl von Wahlanträge als auch Anträge für Ressourcen.

2.6.2. Kurzbeschreibung der Akteure

Antragsverwalter:

Will Anträge bearbeiten und mögliche Fehler berichtigen.

Präsidium:

Will Anträge bearbeiten und mögliche Fehler berichtigen.

2.6.3. Vorbedingungen

1. Benutzer ist angemeldet.
2. Antrag muss eingereicht sein.

2.6.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Akteur offene Antrag im Antragsverwaltungstool öffnet.
2. Akteur prüft, ob Antrag den Regularien entspricht.
3. Akteur speichert Antrag ab/ gibt den Antrag frei.
4. Akteur wählt zugehörigen Antragsstatus.
5. Antrag wird im TOP eingeordnet.
6. Der Use Case ist abgeschlossen.

2.6.5. Wesentliche Szenarios

Erfolgreiche Bearbeitung

Präsidium hat den Antrag nach Regularien entsprechend geändert.

Uneindeutig ausgefüllt

Sc01: Präsidium kann Antrag nicht vollständig bearbeiten, da Aussagen des Antragsstellers uneindeutig sind.

Sc02: Präsidium nimmt Rücksprache zu Antragsteller. Antragssteller muss dann ein Änderungsantrag einreichen.

Sc03: Antrag behält Status der Bearbeitung.

Antragsverwalter bearbeitet Antrag bis es Regularien entspricht

Präsidium kann Antrag so oft bearbeiten bis es alle Kriterien erfüllt.

2.6.6. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

- Alle Angaben entsprechend den Regularien für die entsprechende Antragsstellung
- Antrag wurde von Präsidium verabschiedet
- Antrag wird TOP

2.6.7. Besondere Anforderungen

[System-wide Requirements](#)

2.7. Use-Case 10: Antrag ablehnen

2.7.1. Kurzbeschreibung

Der Use Case beschreibt das Vorgehen zum Ablehnen von Anträgen. Dies kann während der Bearbeitung des Antrags sein sowohl auch bei bearbeitete Anträge während einer Sitzung.

2.7.2. Kurzbeschreibung der Akteure

Antragsverwalter:

Will Anträge bearbeiten und mögliche Fehler berichtigen. Will Antragsstatus verändern.

Präsidium:

Will Anträge bearbeiten und mögliche Fehler berichtigen. Will Antragsstatus verändern.

2.7.3. Vorbedingungen

1. Benutzer ist angemeldet.
2. Antrag muss eingereicht sein und Regularien entsprechen.

2.7.4. Standardablauf (Basic Flow)

1. Akteur öffnet Antrag im Antragsverwaltungstool.
2. Akteur prüft, ob Antrag den Regularien entspricht.
3. Akteur lehnt Antrag ab.
 - a. Akteur gibt eine Begründung ein (Kriterien je nach Antragsart)
4. Antrag wird im TOP als abgelehnt bezeichnet.
5. Der Use Case ist abgeschlossen.

2.7.5. Wesentliche Szenarios

Erfolgreiche Ablehnung

Präsidium hat den Antrag (nach Abstimmung) abgelehnt.

2.7.6. Nachbedingungen

Bei erfolgreicher Durchführung des Use Cases müssen folgende Nachbedingungen erfüllt sein:

- Alle Angaben entsprechend den Regularien für die entsprechende Antragsstellung
- Antrag wurde von Präsidium verabschiedet
- Antrag wird in TO als bearbeitet und abgelehnt bezeichnet
- Antrag von Antragliste entfernt

2.7.7. Besondere Anforderungen

[System-wide Requirements](#)

2.8. Use-Case 11: Antrag vertagen

2.8.1. Kurzbeschreibung

Der Use Case beschreibt das Vorgehen zum Verschieben von Anträgen. Der Befehl kann vor und während einer Sitzung gegeben werden.

2.8.2. Kurzbeschreibung der Akteure

Antragsverwalter:

Möchte Antrag vertagen.

Präsidium:

Möchte Antrag vertagen.

2.8.3. Vorbedingungen

1. Benutzer ist angemeldet.
2. Antrag muss eingereicht sein und Regularien entsprechen.
3. Antrag wurde bereits vom Präsidium bearbeitet und genehmigt.

2.8.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Akteur den Antrag im Antragsverwaltungstool öffnet.
2. Akteur prüft, ob Antrag den Regularien entspricht.
3. Akteur verschiebt Antrag.
4. Antrag wird im TOP eingeordnet.
5. Antrag wird in kommende TO neu aufgelistet.
6. Der Use Case ist abgeschlossen.

2.8.5. Alternative Abläufe

Antrag an TO vertagen

Wenn Akteur im Schritt 1 des Standardablauf die TO aufruft und bearbeitet, dann . kann der Akteur direkt da der Antrag vertagen . Der Use Case wird im Schritt 5 fortgesetzt.

2.8.6. Wesentliche Szenarios

Erfolgreiche Verschiebung

Antragsverwalter hat den Antrag erfolgreich auf nächste TO verschoben.

Sc01: Wenn der Antrag vor der Sitzung vertagt wurde (vor Veröffentlichung der TO), benötigt es eine Begründung.

Sc02: Wenn der Antrag während der Sitzung vertagt wird, benötigt es eine schlüssige Begründung. Dies um die Transparenz der bereits veröffentlichte TO zu garantieren.

Antrag taucht fehlerhaft in nächsten TO ein.

1. Antrag wurde falsch in neue TO eingeordnet, nummeriert und ausgefüllt.
2. Antrag behält Status der Bearbeitung.

Antrag geht verloren

1. Antrag ist nach Verschiebung in nächste TO nicht vorhanden. Es ist aber als offener Antrag noch zu finden.
2. Antrag behält Status der Bearbeitung.

2.8.7. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

Alle Angaben entsprechend den Regularien für die entsprechende Antragsstellung.

1. Antrag wurde von Präsidium verschoben.
2. Antrag wird TOP in neue TO.
3. Antrag behält Bearbeitungsstatus.

2.8.8. Besondere Anforderungen

[System-wide Requirements](#)

2.9. Use-Case 12: Antrag ablehnen

2.9.1. Kurzbeschreibung

Der Use Case beschreibt das Vorgehen zum Ablehnen von Anträgen. Dies kann während der Bearbeitung des Antrags sein sowohl auch bei bearbeitete Anträge während einer Sitzung.

2.9.2. Kurzbeschreibung der Akteure

Antragsverwalter:

Will Anträge bearbeiten und mögliche Fehler berichtigen. Will Antragsstatus verändern.

Präsidium:

Will Anträge bearbeiten und mögliche Fehler berichtigen. Will Antragsstatus verändern.

2.9.3. Vorbedingungen

1. Benutzer ist angemeldet.
2. Antrag muss eingereicht sein und Regularien entsprechen.

2.9.4. Standardablauf (Basic Flow)

1. Akteur öffnet Antrag im Antragsverwaltungstool.
2. Akteur prüft, ob Antrag den Regularien entspricht.
3. Akteur lehnt Antrag ab.
 - a. Akteur gibt eine Begründung ein (Kriterien je nach Antragsart)
4. Antrag wird im TOP als abgelehnt bezeichnet.
5. Der Use Case ist abgeschlossen.

2.9.5. Wesentliche Szenarios

Erfolgreiche Ablehnung

Präsidium hat den Antrag (nach Abstimmung) abgelehnt.

2.9.6. Nachbedingungen

Bei erfolgreicher Durchführung des Use Cases müssen folgende Nachbedingungen erfüllt sein:

- Alle Angaben entsprechend den Regularien für die entsprechende Antragsstellung
- Antrag wurde von Präsidium verabschiedet
- Antrag wird in TO als bearbeitet und abgelehnt bezeichnet
- Antrag von Antragsliste entfernt

2.9.7. Besondere Anforderungen

[System-wide Requirements](#)

2.10. Use-Case 13: Antrag vertagen

2.10.1. Kurzbeschreibung

Der Use Case beschreibt das Vorgehen zum Verschieben von Anträgen. Der Befehl kann vor und während einer Sitzung gegeben werden.

2.10.2. Kurzbeschreibung der Akteure

Antragsverwalter:

Möchte Antrag vertagen.

Präsidium:

Möchte Antrag vertagen.

2.10.3. Vorbedingungen

1. Benutzer ist angemeldet.
2. Antrag muss eingereicht sein und Regularien entsprechen.
3. Antrag wurde bereits vom Präsidium bearbeitet und genehmigt.

2.10.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Akteur den Antrag im Antragsverwaltungstool öffnet.
2. Akteur prüft, ob Antrag den Regularien entspricht.

3. Akteur verschiebt Antrag.
4. Antrag wird im TOP eingeordnet.
5. Antrag wird in kommende TO neu aufgelistet.
6. Der Use Case ist abgeschlossen.

2.10.5. Alternative Abläufe

Antrag an TO vertagen

Wenn Akteur im Schritt 1 des Standardablauf die TO aufruft und bearbeitet, dann . kann der Akteur direkt da der Antrag vertagen . Der Use Case wird im Schritt 5 fortgesetzt.

2.10.6. Wesentliche Szenarios

Erfolgreiche Verschiebung

Antragsverwalter hat den Antrag erfolgreich auf nächste TO verschoben.

Sc01: Wenn der Antrag vor der Sitzung vertagt wurde (vor Veröffentlichung der TO), benötigt es eine Begründung.

Sc02: Wenn der Antrag während der Sitzung vertagt wird, benötigt es eine schlüssige Begründung. Dies um die Transparenz der bereits veröffentlichte TO zu garantieren.

Antrag taucht fehlerhaft in nächsten TO ein.

1. Antrag wurde falsch in neue TO eingeordnet, nummeriert und ausgefüllt.
2. Antrag behält Status der Bearbeitung.

Antrag geht verloren

1. Antrag ist nach Verschiebung in nächste TO nicht vorhanden. Es ist aber als offener Antrag noch zu finden.
2. Antrag behält Status der Bearbeitung.

2.10.7. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case müssen folgende Nachbedingungen erfüllt sein:

Alle Angaben entsprechend den Regularien für die entsprechende Antragsstellung.

1. Antrag wurde von Präsidium verschoben.
2. Antrag wird TOP in neue TO.
3. Antrag behält Bearbeitungsstatus.

2.10.8. Besondere Anforderungen

System-wide Requirements

3. System-Wide Requirements: {Antragsverwaltungstool (I3)}

Richard Frieze <richard.frieze@htw-dresden.de>; Andreas Gröber <andreas.groeber@htw-dresden.de>; Jasmin Henn <jasmin.henn@htw-dresden.de>; Ruben-David Kraus <s80350@htw-dresden.de>; Richard Müller <richard.mueller2@htw-dresden.de>; Duy Tien Nguyen <s80287@htw-dresden.de>; Amanda Wer Estrada <amanda.werestrada@htw-dresden.de>; Jacob Flaskamp <s81424@htw-dresden.de>

3.1. Einführung

In diesem Dokument werden die systemweiten Anforderungen für das Projekt I3 Antragsverwaltungstool spezifiziert. Die Gliederung erfolgt nach der FURPS+ Anforderungsklassifikation:

- Systemweite funktionale Anforderungen (F),
- Qualitätsanforderungen für Benutzbarkeit, Zuverlässigkeit, Effizienz und Wartbarkeit (URPS) sowie
- zusätzliche Anforderungen (+) für technische, rechtliche, organisatorische Randbedingungen

Weiteres unter [Use-Case Model](#).

3.2. Systemweite funktionale Anforderungen

SWFA-1: Für erweiterte Funktionen erfolgt eine Authentifizierung über Login.

Zur Überprüfung wird ein Benutzerkonto durch das Team angelegt und über einen Login-Versuch getestet.

SWFA-2: Ressourcen zur Datenspeicherung sollen gering gehalten werden.

Zur Überprüfung wird zu viel Text in die Maske eingegeben, der durch die Anwendung beschränkt wird.

SWFA-3: Es soll automatisch eine Tagesordnung erzeugt werden

Zur Überprüfung wird aus gestellten Anträgen eine Tagesordnung generiert.

SWFA-4: Automatische Anpassung der TOPs

Zur Überprüfung wird ein Antrag vertagt, wodurch sich seine Priorität erhöht.

3.3. Qualitätsanforderungen für das Gesamtsystem

3.3.1. Benutzbarkeit (Usability)

- Möglichst einfaches Erlernen mit selbsterklärenden Elementen und Beschriftungen, eine Benutzung sollte ohne Vorwissen möglich sein
- Intuitive Bedienung über Nutzerführung durch Text mit einfachen Erläuterungen
- Vorerst wird nur eine deutsche Variante mit deutscher Formatierung entwickelt
- Für die Antragsstellung ist eine Dauer von 10 Minuten vorgesehen

3.3.2. Zuverlässigkeit (Reliability)

- Wartungsarbeiten werden mit den Wartungszeiten des HTW-Servers synchronisiert wenn nötig
- Verfügbarkeit von 90% soll erzielt werden → Maximal ca 2h zum Reparieren
- Netzwerkausfall ist zum jetzigen Zeitpunkt nicht zu kalkulieren
- Sicherung der Anträge soll alle 2h erfolgen (außer im Wartungszustand)
- Komplette Sicherung soll ein mal in der Woche erfolgen, wöchentlich wird das älteste Backup gelöscht um Datensicherheit zu gewährleisten

3.3.3. Effizienz (Performance)

- Priorität der Stura internen Mitarbeiter bei der Benutzung
- Maximal 2 Nutzer in der internen Bearbeitung
- Maximal 10 Nutzer in der externen Antragsstellung
- Performance muss nicht zu hoch ausfallen, da die Anzahl der Anträge gering ist → Fokus auf interne Bearbeitung
- Limitierung der Performance durch Hardware nicht abzuschätzen

3.3.4. Wartbarkeit (Supportability)

- Anpassbarkeit hoch anzusetzen, da neue Funktionen immer einfach eingebunden werden sollen (einfaches Grundgerüst zur Erweiterung)
- Kompatibilität zu Linux/Unix muss gegeben sein, ansonsten browserbasiert → Allumfassend kompatibel zuerst aber nur Desktop
- Apache oder NGINX Webserver
- Skalierbarkeit eher gering da Stura eher geringen Zuwachs an Nutzern verzeichnet
- Konfigurierbarkeit nicht geplant
- Installation nur ein mal auf Server notwendig

3.4. Zusätzliche Anforderungen

3.4.1. Einschränkungen

- Die Software darf nicht unter Java programmiert werden
- Als Webserver wird NGINX genutzt
- Es muss eine Open-Source Anwendung entwickelt werden
- Die Software soll sich optisch an "Antragsgrün" orientieren
- PHP soll möglichst vermieden werden
- Die Zahl an registrierten Nutzern soll gering gehalten werden
- Die Anwendung muss unter LINUX/UNIX lauffähig sein
- Eine vollständige Datenbank soll nicht genutzt werden

3.4.2. Organisatorische Randbedingungen

- Anforderungen an Betrieb, Management und Wartung der Anwendung
- zu beachtende Standards, Normen und Regeln

3.4.3. Rechtliche Anforderungen

- Lizenzierung der Anwendung
- Datenschutz
- Hochschulgesetz

4. Glossar: Antragsverwaltungstool (I3)

Richard Frieze <richard.frieze@htw-dresden.de>; Andreas Gröber <andreas.groeber@htw-dresden.de>; Jasmin Henn <jasmin.henn@htw-dresden.de>; Ruben-David Kraus <s80350@htw-dresden.de>; Richard Müller <richard.mueller2@htw-dresden.de>; Duy Tien Nguyen <s80287@htw-dresden.de>; Amanda Wer Estrada <amanda.werestrada@htw-dresden.de>; Jacob Flaskamp <s81424@htw-dresden.de>

4.1. Einführung

In diesem Dokument werden die wesentlichen Begriffe aus dem Anwendungsgebiet (Fachdomäne) der I3 Antragverwaltungstool-Systems definiert. Zur besseren Übersichtlichkeit sind Begriffe, Abkürzungen und Datendefinitionen gesondert aufgeführt.

4.2. Begriffe

Begriff	Definition und Erläuterung	Synonyme
Änderungsantrag	Formular für die Bearbeitung/Ergänzung/Korrektur eines bereits existierenden und eingereichten Antrages.	(keine)
Antrag	Gesuch, Forderung	Anfrage, Anliegen. Forderung
Antrag (offen)	Antrag, welcher eingereicht wurde, aber noch nicht vollständig vom Präsidium bearbeitet wurde.	offener Antrag
Antragssteller	Jemand, der einen Antrag stellt.	Bewerber, Kandidat, Interessent
Antragsverwalter	Person, die für einen Antrag verantwortlich ist, ihn in Ordnung hält und betreut	Administrator, Manager
Beschluss	[gemeinsam festgelegte] Entscheidung; Ergebnis einer Beratung	Entscheidung, Enschließung, Resolution

Begriff	Definition und Erläuterung	Synonyme
Haushaltsplan	Plan, der der Feststellung und Deckung des Bedarfs an finanziellen Mitteln dient, der zur Erfüllung der staatlichen Aufgaben im Bewilligungszeitraum voraussichtlich notwendig ist. In diesem Fall, die finanzielle Mittel, die dem Stura zur Verfügung stehen.	Finanzplan, Etat
Hochschule für Technik und Wirtschaft Dresden	Bildungsinstitution mit Fokus auf technische und wirtschaftliche Studiengänge	(keine)
Plenum	Vollversammlung einer Körperschaft, besonders der Mitglieder eines Parlaments.	Vollversammlung
Präsidium	Leitendes Gremium einer Versammlung.	Vorsitz, Direktorium, Führerschaft
Referat	Abteilung einer Behörde als Fachgebiet eines Referenten.	Abteilung, Sektion
Referatskollegium	Das Referatskollegium ist die Gesamtheit aller Referate.	(keine)
Referatsleitung	Ein von der Referatsleitung zur Vertretung des Referates im Referatskollegium bestimmtes Mitglied.	(keine)
Ressourcen	Bestand an Geldmitteln, Geldquelle, auf die jemand zurückgreifen kann.	Bestand, Depot
Studentenrat	Eine Form der Studierendenvertretung an Hochschulen.	Studierendenrat
Tagesordnung	Gesamtheit der Themen, die bei einer Sitzung, Versammlung behandelt werden sollen.	Agenda
Tagesordnungspunkt	Punkt einer Tagesordnung.	(keine)
vertagen	eine Sitzung o. Ä. ergebnislos abbrechen und eine weitere Sitzung zu einem späteren Zeitpunkt ansetzen	verschieben

Begriff	Definition und Erläuterung	Synonyme
Vorstand der Geschäftsführung	Sprecherinnen und Sprecher sowie Beauftragte die dem Referatskollegium mit beratender Stimme angehören.	Sprecher/innen

4.3. Abkürzungen und Akronyme

Abkürzung	Bedeutung	Erläuterung
HTW Dresden	Hochschule für Technik und Wirtschaft	Ist eine Fachhochschule in Dresden.
Stura	Studentinnen- und Studentenrat	Ist das zentrale Organ der Studentinnen- und Studentenschaft, u.a. zur Vertretung dieser.
TO	Tagesordnung	Gesamtheit der Themen, die bei einer Sitzung, Versammlung behandelt werden sollen.
TOP	Tagesordnungspunkt	Ist ein Punkt der Tagesordnung
UC	Use Case	funktionale Anforderung eines Systems

4.4. Verzeichnis der Datenstrukturen

Bezeichnung	Definition	Format	Gültigkeitsregeln	Aliase
Anmeldedaten	Zusammensetzung von Benutzername und Passwort.	String	Emailadresse muss @-Zeichen und Punkt enthalten.	Login

5. Domain Model: Antragsverwaltungstool (I3)

5.1. Allgemeine Informationen

Dieses Dokument veranschaulicht die Beziehungen zwischen den einzelnen Akteuren in der Nutzung des Tools und ihre Beziehungen untereinander.

[Domain model] | *Domain_model.jpg*

Projektdokumentation

- Projektplan
- Risikoliste
- Iteration Plan (für zwei ausgewählte Iterationen)

6. Projekt Plan: Antragsverwaltungstool (I3)

6.1. Einleitung

Ziel des Projektes I3 Antragsmodul ist es eine browserbasierte Anwendung für den StuRa der HTW Dresden zu erstellen. Dabei soll das Tool auf die Abläufe des StuRas abgestimmt werden und auf Unix bzw. Linux laufen. Referatsleitungen des StuRas füllen Anträge aus, welchen durch das Antragsverwaltungstool ein Antragsdatum und eine Antragsnummer nach Systematik des StuRa zugewiesen wird. Das Tool soll daraus eine Tagesordnung für kommende Plenumssitzungen generieren.

6.2. Projektorganisation

Unser Projektteam besteht aus den 8 folgenden Personen mit jeweiligen Zuständigkeitsbereichen:

Name	Zuständigkeitsbereich	Backup
Jacob Flaskamp	Implementierung / Test	Tien
Richard Frieze	Analyse	Amanda
Andreas Gröber	Projektmanagement (in SE II) / Filler	Jasmin
Jasmin Henn	Projektmanagement	Andreas
Ruben-David Kraus	Implementierung	Richard
Richard Müller	Entwurf	Ruben-David
Duy Tien Nguyen	Implementierung / Test	Jacob
Amanda Wer Estrada	Analyse	Richard

Das Team trifft sich wöchentlich montags um 17:00 Uhr zu einer Telefonkonferenz auf Discord, in der neue Erkenntnisse, Auswertungen der Kundengespräche und Fragen ausgetauscht werden. Kundenmeetings laufen ebenso über Discord in Form einer Telefonkonferenz ab. Unsere Ansprechpartner sind Axel Schiller und Liselotte Ruthe. An den Kundenmeetings nehmen 3 Mitglieder des Projektteams teil, welche sich abwechseln und die beiden Ansprechpartner. Termine für die Meetings werden auf die Kundenwünsche abgestimmt.

Außerdem wurden innerhalb des Teams kleinere Gruppen gebildet, die sich jeweils mit der Dokumentation der verschiedenen Zuständigkeitsbereiche beschäftigen:

Gruppenmitglieder	Aufgaben
Amanda Wer Estrada, Richard Frieze	Vision, Glossar, Use Case Model
Jasmin Henn, Andreas Gröber	Projektplan, Risikoliste, Iteration Plan
Richard Müller	Architektur-Notizbuch, unterstützt Teams bei: Use Case Model, Iteration Plan

6.3. Projektpraktiken and Messwerte

Als Vorlage für die Projektdokumentation nutzt unser Team OpenUP. Zudem werden unsere Teammeetings protokolliert und anschließend von einer zweiten Person kontrolliert, um alle besprochenen Themen festzuhalten. Kundenmeetings werden aufgezeichnet, sodass alle Teammitglieder die Wünsche der Stakeholder verfolgen können.

6.4. Projektmeilensteine und -ziele

Tabelle 3

It e r a t i o n	Hauptziele	Meilensteine	Erledigt bis
I1	Team kennenlernen, Rollenverteilung, Kommunikationswege festlegen, Kunden kennenlernen, Meetings mit Kunden durchführen, Vorstellung des Kundens erfassen und erste Konzepte entwerfen, Kundengespräche auswerten	1. Teammeeting, 1. Kundenmeeting, 2. Kundenmeeting	20. Dezember 2020
I2	Erstellung der Projektdokumentationen für den Beleg (Projektplan, Risiko-Analyse), Auswertung der Kundengespräche vom Analyse-Team und Erfassung der Daten (Vision, Glossar, Use-Cases), Grundlegende Rahmenbedingungen und Vorstellungen für Implementierung festlegen	3. Kundenmeeting, Projektdokumentationen bearbeiten	10. Januar 2021
I3	Use-Cases mit dem Kunden besprechen und ggf. anpassen, Test Cases erstellen, 4. Kundenmeeting durchführen und auswerten, Team Implementierung beginnt seine Arbeit	4. Kundenmeeting, Beginn Implementierung, Projektdokumentationen fertigstellen	31. Januar 2021

6.5. Bereitstellung der Software

Die fertige Software wird über den HTW Dresden Server laufen. Entweder über den StuRa oder das Rechenzentrum. Installation und Erstinbetriebnahme sind Projektbestandteile. Die Software wird während der Projektlaufzeit entwickelt und innerhalb dieser Zeit das letzte mal geupdatet.

6.6. Gewonnene Erkenntnisse

Ergänzung folgt nach Beendigung des Projektes in SE 2

In diesem Dokument sind die wesentlichen Risiken des Projekts aufgeführt.

6.7. Attribute

Dabei werden folgende Attribute verwendet:

- **Typ:** Ressourcen, Geschäftlich, Technisch, Zeitlich
- **Auswirkung (IMP):** Wert zwischen 1 (niedrig) und 5 (hoch), der die Auswirkungen auf das Projekt angibt, wenn das Risiko eintritt
- **Wahrscheinlichkeit (PRB):** Prozentangabe für die Eintrittswahrscheinlichkeit des Risikos
- **Stärke (MAG):** Produkt aus Auswirkung und Wahrscheinlichkeit (damit kann die Liste sortiert werden)

6.8. Risiken

Die Risiken sind in folgender Tabelle: [Tabelle 1, “Risiken”](#) dargestellt. Das Datum des Dokuments oben gibt an, wann die Risikoliste zuletzt aktualisiert wurde.

Tabelle 1. Risiken

ID	Datum	Name	Beschreibung	Typ	IMP	PRB	MAG	Owner	Gegenmaßnahme
1	01.12.2019	Teammitglied fällt aus	Mitglied des Teams kann dauerhaft nicht mehr am Projekt teilnehmen	z.B. wegen Studiengangswechsels	6	<10%	0,6	Jasmin Henn, Andreas Gröber	Backup-Rollen nutzen
2	01.12.2019	Teammitglied fällt aus	Mitglied des Teams ist zeitweise nicht mehr verfügbar	z.B. technischen Fehlern (Kein Internet, PC/Laptop kaputt) oder Krankheit	3	50%	1,5	Jasmin Henn, Andreas Gröber	Backup-Rollen nutzen und Know-How/Wissen im Team kontinuierlich teilen bzw. weitergeben
3	30.12.2019	Projektaufwand falsch eingeschätzt	Zeitaufwand bzw. Arbeitsaufwand zu klein gewählt	z.B. aufgrund Unerfahrenheit	4	40%	1,6	Jasmin Henn, Andreas Gröber	agile Arbeitsmethoden verwenden z.B. UP (nicht alles an Tag 1 Planen -> Erfahrungen sammeln)
4	30.12.2019	Projektumstände ändern sich	Software kann in der derzeitigen Form nicht mehr verwendet werden aufgrund neuer bzw. aktualisierter Außeneinflüsse	z.B. durch ein neues Gesetz	1-8	<5%	0,05-0,4	Jasmin Henn, Andreas Gröber	-
5	30.12.2019	Stakeholder Interesse/Vorstellung entspricht nicht der eigenen	Misskommunikation mit Stakeholdern bzw. unter den Stakeholdern führt dazu, dass das Produkt abgelehnt wird	z.B. durch falsche Erwartungen oder Vorstellung vom Produkt und fehlender Absprache	4	30%	1,2	Jasmin Henn, Andreas Gröber	Regelmäßige Meetings mit Stakeholdern, Erstellung eines Prototyps

ID	Datum	Name	Beschreibung	Typ	IMP	PRB	MAG	Owner	Gegenmaßnahme
6	30.12.2019	Konflikte im eigenen Team	Konflikte stören den Workflow	durch verschiedene Ansprüche, Interessen oder Zeitaufwand im Bezug auf das Projekt (möglicherweise auch fehlender Erfahrung)	3	20%	0,6	Jasmin Henn, Andreas Gröber	Regelmäßige Meetings in denen eventuelle Probleme geklärt werden können
7	30.12.2019	Probleme im Team durch wechsel der Team-Zusammenstellung (SE I -> SE II)	Wissen geht verloren, (fehlende) Rollen müssen übernommen werden	Wirtschafts-Ingenieure verlassen das Projekt-Team nach SE-I	3	100%	3	Jasmin Henn, Andreas Gröber	Backup-Rollen verteilen, kritische Informationen teilen und festhalten in einer Art Protokoll
8	30.12.2019	Probleme mit Stakeholdern	fehlende Beteiligung, Motivation oder unkonkrete Vorgaben	-	5	<5%	0.25	Jasmin Henn, Andreas Gröber	-
9	30.12.2019	Mangelnde Teamfähigkeit/ Leistung	Einzelne Teammitglieder bzw. das ganze Team leistet nicht genug bzw. schafft dies nicht in eine verwertbare Form zu bringen	z.B durch mangelnde Vorkenntnisse oder mangelnder Erfahrung bzw. fehlender Motivation	5	20%	1	Jasmin Henn, Andreas Gröber	Wissen in dem Teilen, Probleme ansprechen in regelmäßigen Treffen

ID	Datum	Name	Beschreibung	Typ	IMP	PRB	MAG	Owner	Gegenmaßnahme
10	30.12.2019	Fehlende Macht oder Befugnisse	Projekt kann nicht weitergeführt werden durch äußere Einflüsse	z.B fehlende Berechtigung auf die Zugriffsrechte des Htw-Servers bzw. des Rechnernetzwerks der Htw	6-10	??	??	Jasmin Henn, Andreas Gröber	Diese Informationen vorher beschaffen bzw. alternative Lösungen finden
11	30.12.2019	Software ist fehlerhaft	Bugs oder Sicherheitslücken machen Software unbrauchbar	z.B durch Sicherheitslücken die es möglich machen, dass kritische Informationen nach außen dringen	2-6	50%	1,2-3,0	Jasmin Henn, Andreas Gröber	ausreichend testing betreiben

7. Iteration Plan 01: I3_Antragsverwaltungstool

7.1. Meilensteine

Meilenstein	Datum
Beginn der Iteration	17. November 2020
1. Treffen im Discord mit gesamten Team	26. November 2020
1. Kundenmeeting	9. Dezember 2020
2. Kundenmeeting	16. Dezember 2020
Ende der Iteration	20. Dezember 2020

7.2. Wesentliche Ziele

- Team kennenlernen
- Rollenverteilung
- Kommunikationswege festlegen
- Kundenkontakt aufnehmen
- Meetings mit Kunden durchführen
- Vorstellung des Kundens während der Gespräche erfassen und erste Konzepte entwerfen
- Kundengespräche auswerten

7.3. Aufgabenzuordnung

Die folgenden Aufgaben werden in dieser Iteration bearbeitet:

siehe Task List [../task_list.pdf](#)

7.4. Probleme (optional)

Problem	Status	Notizen
einzelne Teammitglieder fehlen	gelöst	Inhalte der Besprechung in Meetingprotokollen festhalten
lange Kommunikationswege	gelöst	feste wöchentliche Termine vereinbaren für Discord-Meetings
kein persönlicher Kontakt zu Teammitgliedern möglich, wegen Covid-19	gelöst	stattdessen Online-Projektarbeit

7.5. Bewertungskriterien

- Anwesenheit bei Meetings
 - Größtenteils sind alle Meeting-Teilnehmer bzw. Stakeholder zu den entsprechenden Meetings in unsrem Discord (Sprach-Chat) anzutreffen. Falls nicht melden sie sich vorher ab.
- Konzeptideen für Software stimmen mit Kundenvorstellungen überein
 - Positives Feedback vom Kunden bezüglich unserer Arbeit erhalten

7.6. Assessment

Assessment Ziel	Das kann die gesamte Iteration oder eine spezifische Komponente sein
Assessment Datum	20. Dezember 2020
Teilnehmer	alle Teammitglieder
Projektstatus	Grün

[I1] | *I1.jpg*

8. Iteration Plan 02: I3_Antragsverwaltungstool

8.1. Meilensteine

Meilenstein	Datum
Beginn der Iteration	21. Dezember 2020
3. Kundenmeeting	6. Januar 2021
Projektdokumentationen bearbeiten	geplant bis 4. Januar 2021
Ende der Iteration	10. Januar 2021

8.2. Wesentliche Ziele

- Erstellung der Projektdokumentationen für den Beleg:
 - Projektplan
 - Risiko-Analyse
- Auswertung der Kundengespräche vom Analyse-Team und Erfassung der Daten:
 - Vision
 - Glossar
 - Use-Cases
- Grundlegende Rahmenbedingungen und Vorstellungen für Implementierung festlegen

8.3. Aufgabenzuordnung

Die folgenden Aufgaben werden in dieser Iteration bearbeitet:

siehe Task List [../task_list.pdf](#)

8.4. Bewertungskriterien

- Übereinstimmung von Kundenmeinung und Vision:
 - Idee bzw. Vorschlag der Vision wurde angenommen
- vorläufige Use-Cases werden vom Kunden bestätigt:
 - Kunde war zufrieden mit Use-Cases, detaillierte Präsentation in Iteration 03

8.5. Assessment

Assessment Ziel	Das kann die gesamte Iteration oder eine spezifische Komponente sein
Assessment Datum	10. Januar 2021
Teilnehmer	alle Teammitglieder
Projektstatus	Grün

[I2+3] | I2+3.jpg

9. Iteration Plan 03: I3_Antragsverwaltungstool

9.1. Meilensteine

Meilenstein	Datum
Beginn der Iteration	11. Januar 2021
4. Kundenmeeting	20. Januar 2021
Beginn Implementierung	über gesamte Iteration
Projektdokumentationen fertigstellen	31. Januar 2021
Ende der Iteration	31. Januar. 2021

9.2. Wesentliche Ziele

- Use-Cases mit dem Kunden besprechen und ggf. anpassen
- Test Cases erstellen
- 4. Kundenmeeting durchführen und auswerten
- Team Implementierung beginnt seine Arbeit

9.3. Aufgabenzuordnung

Die folgenden Aufgaben werden in dieser Iteration bearbeitet:

siehe Task List [../task_list.pdf](#)

9.4. Bewertungskriterien

- Übereinstimmung von Kundenvorstellung und Use-Cases

- positives Feedback zu den Use-Cases erhalten und einige Punkte auf Wunsch des Kunden hin verändert
- Fertigstellung der Projektdokumentationen bis zum Ziel-Termin
 - Zieltermin eingehalten

9.5. Assessment

Assessment Ziel	Das kann die gesamte Iteration oder eine spezifische Komponente sein
Assessment Datum	31. Januar 2021
Teilnehmer	alle Teammitglieder
Projektstatus	Grün

[I2+3] | *I2+3.jpg*

Liste der Anfallenden Aufgaben

Tabelle 2. Zu erledigende Aufgaben

Nr	Aufgabenbeschreibung	Status	Referenzen (Externe Ressourcen)	geplante Iteration	Verantwortlicher
01	Fragen an Kunden für 1. Kundenmeeting zusammenstellen	Done		01	gesamtes Team
02	Kontaktaufnahme zu Felix Müller	Done	Felix Müller	01	Jasmin
03	Termin für 1. Kundenmeeting festlegen und fragen ob Aufzeichnung möglich ist	Done	Axel Schiller	01	Jasmin
04	Aufzeichnung für Kundenmeeting vorbereiten und durchführen	Done		01	Jacob (auch Verantwortlicher für alle kommenden Aufzeichnungen)
05	Termin für 2. Kundenmeeting festlegen	Done	Axel Schiller	01	Jasmin
06	bisherigen Wissenstand zusammenfassen für Kundenmeeting	Done		01	Ruben

Nr	Aufgabenbeschreibung	Status	Referenzen (Externe Ressourcen)	geplante Iteration	Verantwortlicher
07	ersten GUI Entwurf für Kundenmeeting vorbereiten	Done		01	Richard M.
08	Team I4 kontaktieren und nach möglicher Zusammenarbeit erkundigen	Done	Team I4 (Manuela Ziesche)	01	Ruben
09	Vision und Glossar erstellen	Done		02	Amanda und Richard
10	Use-Cases erstellen	Done		02	Amanda, Richard, Richard M.
11	Projektplan und Risikoliste erstellen	Done		02	Jasmin und Andreas
12	Architektur-Notebook erstellen	Done		02	Richard M. und Ruben
13	Termin für 3. Kundenmeeting festlegen	Done	Axel Schiller	02	Jasmin
14	Quellcode der StuRa Website anfordern	Canceled (da nicht notwendig für Implementierung)	Administration Stura Website	02	Tien
15	Grundlegende Methodik für Implementierung festlegen	Done		02	Team Implementierung

Nr	Aufgabenbeschreibung	Status	Referenzen (Externe Ressourcen)	geplante Iteration	Verantwortlicher
16	Fragen an den Kunden zusammenstellen	Done		02	gesamtes Team
17	Use-Cases im Team besprechen	Done		03	gesamtes Team
18	Datenmodell für Datenbank überlegen	Canceled		03	Team Implementierung
19	Use-Cases bearbeiten	Done		03	Richard und Amanda
20	Architecture-Notebook bearbeiten	Done		03	Richard M. und Ruben
21	Vision in Bezug auf Stakeholder abklären	Done		03	Richard
22	Essence Navigator bearbeiten	Done		03	Jasmin
23	Test Cases erstellen	Done		03	Richard M. und Ruben
24	Glossar vervollständigen	Done		03	Richard und Amanda
25	Gewonnene Erkenntnisse in Projektplan ergänzen	Done		03	Jasmin
26	Beleg.pdf dem StuRa zukommen lassen	Done	Axel Schiller	03	Jasmin

Entwurfsdokumentation

- Architektur-Notizbuch
- Test Cases
- Design :leveloffset: +1

Architecture Notebook: Antragsverwaltungstool (I3)

10. Zielsetzung

Dieses Dokument beschreibt die Philosophie, Entscheidungen, Einschränkungen sowie Begründungen der wesentlichen Elemente und alle anderen übergreifenden Aspekte des Systems, die das Design und die Implementierung beeinflussen.

11. Entwurfsziel und Philosophie

Das Programm soll das Einreichen sowie das Ver- und Bearbeiten von Anträgen vereinfachen und automatisieren. Da die Endbenutzer keine einheitliche Struktur bei der Wahl der Hard- und Software haben, ist diese Software browserbasiert, was sie im weitesten Sinne plattformunabhängig macht. Das System soll wenn möglich auf lange Zeit im Stura genutzt werden und wenn möglich auch Platz für Erweiterungen bieten. Vorerst sind keine schnellen Änderungen der Software durch Add-ons oder neue Soft- sowie Hardware zu erwarten, dennoch ist hier eine einfache Erweiterung erstrebenswert.

12. Annahmen und Abhängigkeiten

- Webserver NGNX oder Apache, bereits verwendet und Wissen auf Kundenseite vorhanden
- LINUX/UNIX als Betriebssystem daher Entwicklung für diese Systeme
- Server von Rechenzentrum
- Alle Technologien verwendbar, kein closed Source

13. Architektonisch bedeutende Anforderungen (ergibt sich aus Use-Cases und Furbs+)

- Kein Java darf verwendet werden (Von Kunde vorgeschrieben)
- PHP möglichst vermeiden (empfohlen)
- Freie Software, da nicht kommerziell
- Aussehen ähnlich Antragsgrün
- Nutzerzahl soll gering gehalten werden (niedriger Traffic)
- Lauffähig unter UNIX/LINUX
- Bekanntes Webframework sinnvoll
- Datenschutz Konformität mit Anmeldungen

14. Entscheidungen, Einschränkungen und Begründungen (Programmiersprachen ,Bibliotheken, Frameworks, DBS- Managementsystem usw + Begründung)

- Javascript, sehr weit verbreitet & Vorwissen vorhanden
- CSS daher auch notwendig
- HTML für Webseite
- Webserver entweder NGNX oder Apache da vorhanden im Stura/HTW
- Datenbank möglich, aber eher nicht sinnvoll da Nutzerzahl nicht so hoch skaliert

15. Architekturmechanismen (Technische Umsetzung die sich aus Anforderung ergibt, aus Anforderungen welche Technik ich brauche)

- Log-in mit Oberfläche und backend
- Generieren einer Tagesordnung
- Sortierung der Tagesordnung
- Zwischenspeicherung der eingereichten Anträge

16. Abstraktion

Die Aufgabe des Systems ist es, das Einreichen von Anträgen für Außenstehende und Mitglieder des Sturas zu vereinfachen. Dabei soll eine innovative Oberfläche helfen. Das System generiert ebenfalls eine individuelle Tagesordnung für das Präsidium, welches über die eingereichten Anträge entscheidet und abstimmt. Hier wird mit einer Hierarchie gearbeitet, welche die passende Reinform der eingereichten Anträge gemessen an ihrer Wichtigkeit gewährleistet. Nach dem Verabschieden eines Antrages wird dieser in einer PDF-Datei abgespeichert.

17. Architektonischer Rahmen

- Kompatibilität zu allen gängigen Webbrowsern wie Chrome, Firefox, Safari etc.
- Verwendung von HTML und CSS für die Oberfläche
- Javascript und falls nötig Python für die Mechaniken der Website
- Konfigurieren des Linux/Unix Servers/Webservers als Vorbild soll die Struktur der HTW Webserver dienen
- Generierung der Tagesordnung

18. Architektonische Ansichten

Die Generierung der individuellen Tagesordnung ist das Herzstück des Systems und wird viel Zeit in Anspruch nehmen. Das Zwischenspeichern und Aufarbeiten der eingereichten Anträge für die Tagesordnung ebenfalls.

19. Test Cases: Antragsverwaltungstool (I3)

Richard Frieze <richard.frieze@htw-dresden.de>; Andreas Gröber <andreas.groeber@htw-dresden.de>; Jasmin Henn <jasmin.henn@htw-dresden.de>; Ruben-David Kraus <s80350@htw-dresden.de>; Richard Müller <richard.mueller2@htw.dresden.de>; Duy Tien Nguyen <s80287@htw-dresden.de>; Amanda Wer Estrada <amanda.werestrada@htw-dresden.de>; Jacob Flaskamp <s81424@htw-dresden.de>

19.1. Allgemeine Informationen

19.1.1. allgemeine Tests:

- Abfangen ungültiger Eingaben (z.B. Datum)
- korrekte Übertragung der Daten nach Erstellung/Änderung eines Antrages
- Zugriffsberechtigungen für verschiedene Funktionen der Software
- generelle Funktionalität aller beabsichtigten Funktionen
- korrekte Handhabung unerwarteter Fehler
- korrekte Erstellung des TOP, nachdem neuer Antrag gestellt wurde

19.2. Test Cases

19.2.1. Wahlantrag stellen

- Einreichen nur bei vollständig ausgefülltem Antrag möglich

19.2.2. Ressourcenantrag stellen

19.2.3. offene Anträge ansehen

- korrekte Filterung der gesuchten Anträge

- Aufrufen eines angezeigten Antrages
- direkte Möglichkeit zur Bearbeitung (im internen Bereich)

19.2.4. Tagesordnung ansehen

1. als externer Nutzer

19.2.5. Änderungsantrag