

Anforderungsspezifikation

Inhaltsverzeichnis

1. Vision	1
1.1. Einführung	1
1.2. Positionierung	1
1.3. Stakeholder Beschreibungen	3
1.4. Produkt-/Lösungsüberblick	4
2. Use-Cases	6
2.1. Allgemeine Informationen	6
2.2. Identifizierte Use Cases	6
2.3. Use-Case 01: Kandidaten verwalten	8
2.4. Use-Case 02: Mitglieder aufnehmen	11
2.5. Use-Case 03: Mitgliederinformationen einpflegen	13
2.6. Use-Case 04: Mitglieder per E-Mail kontaktieren	16
3. System-Wide Requirements	17
3.1. Einführung	17
3.2. Systemweite funktionale Anforderungen	17
3.3. Qualitätsanforderungen für das Gesamtsystem	17
3.4. Zusätzliche Anforderungen	17
4. Glossar	19
4.1. Einführung	19
4.2. Begriffe	19
4.3. Abkürzungen und Akronyme	19
4.4. Datenstrukturen	20
5. Domänenmodell	21

Chapter 1. Vision

1.1. Einführung

Der Zweck dieses Dokuments ist es, den wesentlichen Bedarf und Funktionalitäten der Weiterentwicklung der Mitgliederdatenbanken zu sammeln, zu analysieren und zu definieren. Der Fokus liegt auf den Fähigkeiten, die von Stakeholdern und adressierten Nutzern benötigt werden und der Begründung des Bedarfs. Die Details, wie die Weiterentwicklung der Mitgliederdatenbank diesen Bedarf erfüllt, werden in den Use-Cases beschrieben.

1.1.1. Zweck

Der Zweck dieses Dokumentes 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 die Weiterentwicklung der Mitgliederdatenbank, die von der SE/RE I4-Gruppe entwickelt wird. Das System wird es dem Studentenrat erlauben, seine Mitgliederdatenbank optimal darzustellen, um eine bessere Benutzerfreundlichkeit zu erreichen.

1.1.3. Definitionen, Akronyme und Abkürzungen

siehe Glossar.

1.2. Positionierung

1.2.1. Fachliche Motivation

Im Rahmen des Moduls 'Software Engineering', welches sich über zwei Semester erstreckt, müssen Studierende verschiedener Studiengänge in Gruppen ein Softwareprojekt erarbeiten, welches von richtigen Kunden (Themenstellern) genutzt werden soll. Unser Team I4 sollte sich mit der Weiterentwicklung der Mitgliederdatenbank des Studentenrates (StuRa) der HTW Dresden beschäftigen.

Aktuell benutzen die Mitglieder des Studentenrates Excel-Tabellen für die Mitglieder- und Kandidatenverwaltung. Mitglieder und Kandidaten werden dort neu angelegt, verwaltet und gelöscht. Außerdem wird in einer weiteren Excel-Tabelle die Aufwandsentschädigung berechnet. Diese sind durch die häufige Überarbeitung verschiedener Mitglieder unübersichtlich geworden und sind der heutigen Zeit nicht mehr gerecht. Aus diesem Grund soll die vom letzten Jahr entwickelte Mitgliederdatenbank von unserem Team I4 weiterentwickelt und optimiert werden, sodass nun auch Kandidaten angelegt und bearbeitet werden können, diese einfach zu Mitgliedern umgewandelt werden können sobald sie gewählt worden sind und Mitglieder andere Mitglieder per E-Mail kontaktieren können. Desweiteren soll der Arbeitsaufwand des Admins mittels Checkboxes verringert werden, die für mehr Übersichtlichkeit seiner Aufgaben sorgen sollen.

Allgemein sollen die Abläufe der Mitgliederdatenbank soweit optimiert werden, dass alle

Beteiligten einen geringeren Zeitaufwand haben. Durch die intuitive Benutzeroberfläche erhoffen wir uns außerdem, dass es zu einer erleichterten Interaktion mit den Daten kommt.

1.2.2. Problem Statement

Das Problem	Unübersichtliche und unpraktische Art die Daten in Mitgliederdatenbank anzulegen/einzupflegen
betrifft	Mitglieder des StuRa
die Auswirkung davon ist	lange und fehleranfällige Bearbeitung der Daten
eine erfolgreiche Lösung wäre	Bearbeitung der vorhandenen Software, sodass die Prozesse beschleunigt werden und die Bedienbarkeit verbessert wird

1.2.3. Positionierung des Produkts

Für	Mitglieder des Studentenrates
deren	Daten effizienter verwaltet werden können
Das Produkt / die Lösung ist eine	Webseite
die	die essenziellen Daten für die Mitglieder aufbereitet und zuordnet
Im Gegensatz zu	Excel-Tabellen
Unser Produkt	stellt nur die für den Nutzer relevanten Informationen komfortabel dar

Für	den Laborbereich der HTW Dresden
deren	Server für die Datenbank des StuRa verwendet werden.
Das Produkt / die Lösung	Webseite
die	die essenziellen Daten für die Mitglieder aufbereitet und zuordnet
Im Gegensatz zu	Excel-Tabellen
Unser Produkt	stellt nur die für den Nutzer relevanten Informationen komfortabel dar

Für	den Admin
der	
Das Produkt / die Lösung	am meisten nutzen wird.
die	die essenziellen Daten für die Mitglieder aufbereitet und zuordnet
Im Gegensatz zu	Excel-Tabellen
Unser Produkt	stellt nur die für den Nutzer relevanten Informationen komfortabel dar

1.3. Stakeholder Beschreibungen

1.3.1. Zusammenfassung der Stakeholder

Name	Beschreibung	Verantwortlichkeiten
HTW	Hochschule	stellt Studenten für StuRa, zahlt AE an die jeweiligen Studenten
Mitglieder des StuRa	Repräsentanten der Studierenden	Nutzer des Produkts
Systembetreuer, Admin	Systemadministrator	Sind für die Funktionalität, Wartung und Aktualität der Mitgliederdatenbank verantwortlich
Herr Professor Anke	Dozent	Betreut das Projekt und das Team I4, Ansprechpartner für das Team I4
Team I4	Gruppe von Studierenden, die das Projekt übernehmen	Versuchen die Wünsche der Themensteller umzusetzen (das Projekt bearbeiten)
Laborbereich	Laborbereich der Hochschule	Diese stellen den Server, worauf die Datenbank laufen soll
Themensteller	unsere Aufgabensteller	Deren Ziel ist es, eine gut funktionierende und qualitativ hochwertige Datenbank zu erhalten.
Hacker	Jemand, der illegal in Computersysteme eindringt	Stellt Gefahr für das System dar: Fehlfunktionen, Datendiebstahl
der Gesetzgeber	Gibt rechtliche Rahmenbedingungen vor, z.B. durch Gesetze für Jugendschutz, Datenschutz und Fernabsatz	überwacht Gesetze und Regelungen hinsichtlich der Einhaltung des Telemediengesetzes

1.3.2. Benutzerumgebung

1. Anzahl der Personen, die an der Erfüllung der Aufgabe beteiligt sind, ändert sich mit der Zeit (voraussichtlich) nicht.
2. Die Bearbeitung der Administrationsaufgaben (siehe Glossar) dauert unter 3 Min.
3. Es muss gewährleistet werden, dass 5 Benutzer gleichzeitig mit dem Programm arbeiten können.
4. Besondere Umgebungsbedingungen:
 - Die Weiterentwicklung der Mitgliederdatenbank muss weiterhin eine responsive Webseite gewährleisten, damit die Bedienung auch unterwegs mit dem Handy funktioniert.
 - Die Webseite muss zu den Kernarbeitszeiten online sein.
5. Diese Systemplattformen werden zukünftig weiterhin eingesetzt: Windows, Linux, iOS,

Android.

6. Thunderbird muss zur E-Mail Nutzung integriert werden.

1.4. Produkt-/Lösungsüberblick

1.4.1. Bedarf und Hauptfunktionen

Bedarf	Priorität	Features	Geplante s Release
einfache Verwaltung der Kandidaten	Hoch	eigener "Kandidaten" Tab, in dem die Daten des Kandidaten (Name, Vorname, Wahldatum, E-Mail, Beschlussnummer) angelegt und bearbeitet werden können. Kandidaten können auch gelöscht werden. Zusätzlich können relevante Informationen gepflegt werden.	KW26
Aufgaben können übersichtlich abgearbeitet werden	Hoch	für Admin des StuRa werden einzelne Aufgaben automatisch nach Mitgliedsaufnahme angelegt, welche abgehakt werden können.	KW26
vertrauliche Informationen können nur von Admins eingesehen werden	Hoch	"Kandidaten" Tab darf nur von Admins und nicht von Mitgliedern gesehen werden; Telefonnummer von anderen Mitgliedern dürfen auch nur Admins angezeigt werden	KW26
einfaches Mittel zur (Gruppen)-Kommunikation	Mittel	Mailverteiler oder Direktmail mittels Einbindung von Thunderbird	KW26
Automatisierung der Mitgliederaufnahme nach der Wahl	Mittel	Übertragung des Kandidaten zum Mitglied	KW26
Automatische Stimmzettelerstellung	Mittel	Stimmzettel eventuell ausdrucken	—
Workload soll hinzugefügt werden	Niedrig	—	—
Aufwandsentschädigungszahlung vereinfachen	Niedrig	—	—
Organigramm aktualisieren	Niedrig	das existierende Organigramm übersichtlicher gestalten	—

Anforderung	Priorität
System muss auf allen gängigen Browsern sowie auf mobilen Endgeräten lauffähig sein	Hoch

Anforderung	Priorität
Einfache Bedienbarkeit	Mittel
System kann nur online genutzt werden (nicht offline)	Mittel

Chapter 2. Use-Cases

2.1. Allgemeine Informationen

Akteur	Ziel
Admin des Studentenrats	Einpfelegen der Mitglieder und Kandidaturen und Abarbeitung der Check-Listen
Mitglieder StuRa	Übersichtliche Einsehbarkeit von Informationen von anderen Mitgliedern

2.2. Identifizierte Use Cases

Nr	Bezeichnung	Priorität	Kurzbeschreibung
01	Kandidaten verwalten	hoch	Kandidaturen sollen angelegt, bearbeitet oder entfernt werden können.
02	Mitglieder aufnehmen	hoch	Die kandidierende Person wurde gewählt, ist nun Mitglied und soll unter dieser Bezeichnung im System zu finden sein.
03	Mitgliederinformationen einpflegen	hoch	Mitgliederdaten sollen ergänzt werden können.
04	Mitglieder per E-Mail kontaktieren	mittel	Mitglieder sollen andere Mitglieder per Mail kontaktieren können.
06	Workload erfassen	gering	Das Maß der Aktivität der Mitglieder soll vom System erfasst werden können.
07	Aufwandsentschädigung berechnen	gering	Anhand seiner Aktivität soll die Berechnung der Aufwandsentschädigung für das Mitglied erfolgen.
05	Stimmzettel generieren	sehr gering	Generierung von Stimmzetteln (als druckbares Dokument) mit allen Kandidaturen, die sich zu einer Wahl aufstellen lassen.


```

@startuml diagramm_use_case-model
left to right direction

actor Admin
actor Mitglied

rectangle "Webseite des STURA" {
    "UC 01: Kandidaten verwalten" as (uc01) #A9E2F3
    "UC 02: Mitglieder aufnehmen" as (uc02) #A9E2F3
    "UC 03: Mitgliederinformationen einpflegen" as (uc03) #A9E2F3
    "UC 04: Mitglieder per E-Mail kontaktieren" as (uc04) #A9E2F3
    "UC 05: Stimmzettel generieren" as (uc05) #A9E2F3
    "UC 06: Workload erfassen" as (uc06) #A9E2F3
    "UC 07: Aufwandsentschädigung berechnen" as (uc07) #A9E2F3
}

Admin -- (uc01)
Admin -- (uc02)
Admin -- (uc03)
Admin -- (uc04)
Mitglied -- (uc04)
Admin -- (uc05)
Admin -- (uc06)
Admin -- (uc07)

@enduml

```

2.3. Use-Case 01: Kandidaten verwalten

2.3.1. Kurzbeschreibung

Das System ermöglicht es Kandidaturen anzulegen, zu bearbeiten oder zu entfernen.

2.3.2. Kurzbeschreibung der Akteure

Admin

Ist für die Verwaltung zuständig

2.3.3. Vorbedingungen

- Die Internetseite ist geöffnet
- Der Admin muss angemeldet sein

2.3.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Admin den Tab "Kandidaturen" auswählt
2. Der Admin entscheidet sich dazu eine neue Kandidatur anzulegen und teilt dies dem System mit
3. Daraufhin kann der Admin eine neue Kandidatur anlegen, indem er die Daten der kandidierenden Person dem System übergibt
4. Der Use Case ist abgeschlossen, sobald die Eingaben abgespeichert wurden.



Wireframes

Das folgende Wireframe zeigt einen ersten Entwurf der Benutzeroberfläche, die die Anforderungen, die bis zu diesem Zeitpunkt gestellt worden sind, umsetzt. Eine Abweichung zum aktuellen Use Case ist somit möglich.

Kandidaturen

Vorname Nachname Nickname

Funktionen

Organisationseinheit

Organisationseinheit auswählen



+ FUNKTION HINZUFÜGEN

Wahldatum

Beschlusnummer

E-Mail-Adresse

E-Mail



📁 SPEICHERN

ABBRECHEN

Figure 1. Wireframe Kandidatur hinzufügen

2.3.5. Alternative Abläufe

Alternativer Ablauf 1

1. Der Admin möchte eine Kandidatur bearbeiten.
2. Der Admin wählt die Möglichkeit der Bearbeitung der zu bearbeitenden Kandidatur aus
3. In der aufgerufenen Maske können die Änderungen vorgenommen werden.
4. Die Änderungen werden gespeichert.

Alternativer Ablauf 2

1. Der Admin hat nicht alle geforderten Daten angegeben

2. Der Vorgang der Abspeicherung schlägt fehl
3. Dem Admin wird eine entsprechende Fehlermeldung ausgegeben

Alternativer Ablauf 3

1. Der Admin hat die Intention eine Kandidatur zu löschen
2. Für die jeweilige Kandidatur wählt er diese Option aus, woraufhin sich der Löschvorgang einleitet

2.3.6. Nachbedingungen

Nach der Einpflegung einer neuen Kandidatur, erscheint ein Teil dieser Daten in der Übersicht aller gelisteten Kandidaturen

2.3.7. Hinweise

Ablauf soll sich an Mitglieder Tab orientieren

2.4. Use-Case 02: Mitglieder aufnehmen

2.4.1. Kurzbeschreibung

Die kandidierende Person wird als Mitglied gewählt und im System als solches übernommen.

2.4.2. Kurzbeschreibung der Akteure

Admin

Ist für die Mitgliederaufnahme und die Bearbeitung der Check-Listen zuständig

2.4.3. Vorbedingungen

- Die Internetseite ist geöffnet
- Der Admin muss angemeldet sein
- Die kandidierende Person wurde laut den Wahlen gewählt
- Sie hat der Übernahme als Mitglied zugestimmt und hat alle erforderlichen Dokumente unterschrieben dem Admin zukommen lassen
- Check-Listen-Template muss für jede Organisationseinheit vorhanden sein

2.4.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Admin den Tab "Kandidaturen" auswählt
2. Der Admin wählt die Möglichkeit aus, die kandidierende Person als Mitglied aufzunehmen
3. Der Datensatz der Kandidatur wird aus der Übersicht entfernt und taucht unter dem Tab "Mitglieder" auf, in der er bearbeitet werden kann



Wireframes

Das folgende Wireframe zeigt einen ersten Entwurf der Benutzeroberfläche, die die Anforderungen, die bis zu diesem Zeitpunkt gestellt worden sind, umsetzt. Eine Abweichung zum aktuellen Use Case ist somit möglich.

Kandidaturen

Suchen...



+ HINZUFÜGEN

ENTFERNEN

Name	Funktionen	E-Mails		
<input type="checkbox"/> Axel Schiller	Stellvertretung Finanzerinnen und Finanzer (Fachausschuss Design)	schiller@stura.htw-dresden.de	+ Aufnahme	✎
<input type="checkbox"/> max-mar Mustermann	Service/Buchhaltung (Angestellte)	max.mustermann@stura-dresden-htw.de	+ Aufnahme	✎
<input type="checkbox"/> TEst Test II	Mitglied als Antrags- und Beschlussverwaltung (Präsidium)	Tester@stura.htw-dresden.de	+ Aufnahme	✎
<input type="checkbox"/> Tester Test	-	Tester@stura.htw-dresden.de	+ Aufnahme	✎

1

Figure 2. Wireframe Kandidatur zu Mitglied aufnehmen

2.4.5. Nachbedingungen

- Die Kandidatur wurde zum Mitglied umgewandelt und kann unter dem Tab "Mitglieder" bearbeitet werden

2.5. Use-Case 03: Mitgliederinformationen einpflegen

2.5.1. Kurzbeschreibung

Administrator kann weitere Informationen für neu angelegte Mitglieder einpflegen.

2.5.2. Kurzbeschreibung der Akteure

Admin

Ist für die Aufnahme und Bearbeitung der zusätzlichen Informationen zuständig

2.5.3. Vorbedingungen

- Die Internetseite ist geöffnet
- Der Admin muss angemeldet sein
- Die unterschriebenen Dokumente und Einwilligung bzw. der Widerspruch wurden von den neuen Mitgliedern dem Admin zugesendet.

2.5.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Admin den Tab "Mitglieder" auswählt
2. Der Admin wählt ein Mitglied, dass bearbeitet werden muss aus
3. Eine Checkliste gibt Auskunft über noch offene Aufgaben, die der Admin bearbeiten muss
 - a. unterschriebene Dokumente hochladen (Verpflichtung auf das Datengeheimnis, Kenntnisnahme der Ordnungen, Stammdatenblatt)
4. Er gibt an, ob die Telefonnummer des neuen Mitgliedes im Notfall an andere Mitglieder weitergegeben werden darf




Wireframes

Die folgenden Wireframes zeigen einen ersten Entwurf der Benutzeroberfläche, die die Anforderungen, die bis zu diesem Zeitpunkt gestellt worden sind, umsetzten. Eine Abweichung zum aktuellen Use Case ist somit möglich.

Mitglied bearbeiten

Vorname Nachname Nickname

Funktionen

Organisationseinheit Bereich Funktion 

Beginn Amtszeit Ende Amtszeit

Organisationseinheit Bereich Funktion 

Beginn Amtszeit Ende Amtszeit

[+ FUNKTION HINZUFÜGEN](#)

E-Mail-Adressen

E-Mail 

[+ E-MAIL HINZUFÜGEN](#)

Anschrift und Kontaktdaten

Straße Hausnummer

Postleitzahl Ort

Telefon (mobil)

Dokument 1 

[+ Upload](#)

 **SPEICHERN**

ABBRECHEN

Figure 3. Wireframe Mitglied bearbeiten

Kandidaturen

- ☐ Wahl angenommen
- ☐ Kenntnisnahme der Ordnungen
- ☐ Verpflichtung auf das Datengeheimnis
- ☐ Stammdatenblatt

Aufnehmen **SPEICHERN**

Figure 4. Wireframe Checkliste

2.5.5. Alternative Abläufe

Alternativer Ablauf 1

1. wenn der Admin eine zu große Datei hochlädt, kommt eine Fehlermeldung

2.5.6. Nachbedingungen

1. aktualisierte Informationen wurden hinterlegt

2.5.7. Besondere Anforderungen

- einfache Bedienbarkeit ermöglichen
- Hochladen der Informationen muss in dazu angemessener Zeit erfolgen

2.6. Use-Case 04: Mitglieder per E-Mail kontaktieren

2.6.1. Kurzbeschreibung

Über die Mitgliederdatenbank ist es angemeldeten Mitgliedern möglich, via E-Mail Kontakt zu anderen Mitgliedern herzustellen

2.6.2. Kurzbeschreibung der Akteure

Mitglied des Stura

Mitglieder des Stura sind in der Lage die, auf der Webseite der Mitgliederdatenbank, dargestellten Informationen einzusehen, die für sie freigegeben sind.

2.6.3. Vorbedingungen

- Die Internetseite ist geöffnet
- Das Mitglied muss angemeldet sein

2.6.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn sich ein Mitglied dazu entscheidet Kontakt zu einem anderen Mitglied aufnehmen zu wollen
2. Dazu wird der "Mitglieder"-Tab ausgewählt, der eine gelistete Übersicht über alle Mitglieder enthält, die sich zu diesem Zeitpunkt in der Datenbank befinden
3. Anschließend wird das gewünschte Mitglied rausgesucht, zu dem der Kontakt aufgebaut werden soll
4. Die angegebene Mail-Adresse kann nun eingesehen werden
5. Unter Kontaktdaten befindet sich neben der E-Mail-Adresse, in schriftlicher Form, auch eine Option, die für die direkte Kontaktierung vorgesehen ist
6. Der Use Case ist abgeschlossen, wenn sich das Mitglied dazu entscheidet die ausgelesene E-Mail-Adresse für die Kontaktierung zu verwenden, oder die Option zur direkten Kontaktaufnahme betätigt wurde (infolgedessen öffnet sich ein externes Mailprogramm mit einer neuen E-Mail, welches die Mail-Adresse des ausgewählten Mitgliedes schon im Empfängerfeld stehen hat)

2.6.5. Alternative Abläufe

Alternativer Ablauf 1

1. Ist das Mitglied, das kontaktiert werden möchte, nicht in der Übersicht zu finden, ist der Use Case vorzeitig abgeschlossen (in diesem Fall existieren keine Daten zur gewünschten Person in der Datenbank und eine Kontaktaufnahme ist nicht möglich)

2.6.6. Besondere Anforderungen

- ermöglicht eine zeitsparende Kontaktaufnahme

Chapter 3. System-Wide Requirements

3.1. Einführung

In diesem Dokument werden die systemweiten Anforderungen für das Projekt I4 Weiterentwicklung der Mitgliederdatenbank des StuRa 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

3.2. Systemweite funktionale Anforderungen

Ausarbeitung eines Datenschutzkonzeptes, welches die Anforderungen jederzeit erfüllt:

F1: Die personenbezogenen Daten sollen vor unberechtigtem Zugriff geschützt sein.

F2: Die Daten sollen persistent sein.

F3: Die eingegebenen Werte sollen validiert bzw. der Wertebereich geprüft werden.

3.3. Qualitätsanforderungen für das Gesamtsystem

3.3.1. Benutzbarkeit (Usability)

U1: Das Mitglied/ Der Admin sollte die Bedienung innerhalb eines Tages erlernen.

U2: Ausführen einer Hauptfunktionalität darf nicht mehr als 10 Klicks beanspruchen.

3.3.2. Zuverlässigkeit (Reliability)

R1: Datenbank soll einmal die Woche ein Backup machen. Maximal drei Versionen der Datenbank sollen gespeichert werden.

3.3.3. Effizienz (Performance)

P1: Die Ladezeiten einer neuen Seite sollte sich auf zwei Sekunden beschränken, unter der Voraussetzung einer 16Mbit/s-Anschlusses.

3.3.4. Wartbarkeit (Supportability)

S1: Das System soll einfach erweiterbar und wartbar sein.

3.4. Zusätzliche Anforderungen

3.4.1. Einschränkungen

- Ressourcenbegrenzungen - Speicherplatzbegrenzung auf 5MB pro Dokument, welches hochgeladen wird.
- Datenbank soll dreimal die Woche ein Backup machen. Maximal drei Versionen der Datenbank sollen gespeichert werden.

3.4.2. Rechtliche Anforderungen

- Nutzung von freien Lizenzen (Open Source)

Chapter 4. Glossar

4.1. Einführung

In diesem Dokument werden die wesentlichen Begriffe aus dem Anwendungsgebiet (Fachdomäne) der Weiterentwicklung der Mitgliederdatenbank des StuRa definiert. Zur besseren Übersichtlichkeit sind Begriffe, Abkürzungen und Datendefinitionen gesondert aufgeführt.

4.2. Begriffe

Begriff	Definition und Erläuterung	Synonyme
Beschlusnummer	wird jedem Antrag im StuRa zugeordnet	—
Studentenrat	Studentische Vertretung an der HTW Dresden	StuRa
Nutzer:in	hat Zugriff auf das StuRa System	User
Mitglied	Mitglied des StuRa	—
Kandidatur	Person, die sich zu Wahl gestellt hat um Mitglied im StuRa zu werden	Kandidat:in
Admin	Zuständig für die administrative Aufgaben der Webseite	Systemverwalter
Check-Liste	Liste, die Aufgaben enthält, die der Admin erledigen muss	To-do-Liste, Erledigungs-Liste
Aufwandsentschädigung	Bezahlung für die Abarbeitung der Aufgaben	—
Workload	Maß der Aktivität des Mitgliedes	Fortschritt
Organisationseinheit	Der StuRa ist in mehrere Organisationseinheiten aufgeteilt, welche dann noch in Bereiche und einzelne Funktionen unterteilt sind	Abteile des StuRa
Dokumente der Kandidaten	Verpflichtung auf das Datengeheimnis, Kenntnisnahme der Ordnungen, Stammdaten	—
Aufgaben des Admins	Kandidaten einpflegen, Kandidaten als Mitglieder aufnehmen, Aufwandsentschädigung berechnen, Mitgliederdaten bearbeiten/ergänzen	

4.3. Abkürzungen und Akronyme

Abkürzung	Bedeutung	Erläuterung
AE	Aufwandsentschädigung	siehe oben
StuRa	Studentenrat	

Abkürzung	Bedeutung	Erläuterung
HTW	Hochschule für Technik und Wirtschaft Dresden	

4.4. Datenstrukturen

Bezeichnung	Definition	Format	Gültigkeitsregeln	Aliase
Kandidatendaten	Vorname, Name, Spitzname, E-Mail, Beschlussnummer, Wahldatum	string, date	Wahldatum darf nicht in der Vergangenheit liegen	Daten des Kandidaten/ der Kandidatin/ der Kandidatur/ der kandidierenden Person
Mitgliederdaten	Vorname, Name, Spitzname, E-Mail, Anschrift, Telefonnummer, Funktion, Eintrittsdatum	string, date		Daten des Mitgliedes
Anschrift	Straße, Hausnummer, Postleitzahl, Ort, Land	string		Adresse

Chapter 5. Domänenmodell

