

Anforderungsspezifikation

Mitgliederdatenbank für den StuRa (I06)

Paul Wentzel, Georg Schicker, Nathalie Kästner, Collin Neumann, Lennart
Bronke

26. Juni 2023

Inhaltsverzeichnis

1. Vision: Mitgliederdatenbank für den StuRa (I06)	1
1.1. Einführung	1
1.2. Positionierung	1
1.3. Stakeholder Beschreibungen	2
1.4. Produkt-/Lösungsüberblick	3
1.5. Zusätzliche Produktanforderungen	4
2. Use-Case Model: Mitgliederdatenbank für den StuRa (I06)	5
2.1. Allgemeine Informationen	5
2.2. Identifizierte Use Cases	5
2.3. Use Case Diagramm	5
2.4. Use-Case: UC01 Mitglied anlegen	7
2.5. Use-Case: UC02 Mitgliederdaten bearbeiten	9
2.6. Use-Case: UC03 Historie ausgeben	11
2.7. Use-Case: UC04 Logdateien erstellen	12
2.8. Use-Case: UC05 Daten importieren	13
2.9. Use-Case: UC06 Daten exportieren	14
2.10. Use-Case: UC07 Checkliste erstellen	15
2.11. Wireframes	15
3. System-Wide Requirements: Mitgliederdatenbank für den StuRa (I06)	19
3.1. Einführung	19
3.2. Systemweite funktionale Anforderungen	19
3.3. Qualitätsanforderungen für das Gesamtsystem	19
3.4. Zusätzliche Anforderungen	21
4. Glossar: Mitgliederdatenbank für den StuRa (I06)	22
4.1. Einführung	22
4.2. Begriffe	22
4.3. Abkürzungen und Akronyme	23
4.4. Verzeichnis der Datenstrukturen	23
5. Domain Model: Mitgliederdatenbank für den StuRa (I06)	24

1. Vision: Mitgliederdatenbank für den StuRa (I06)

1.1. Einführung

Ziel dieses Dokuments ist es, die wesentlichen Bedarfe und Funktionalitäten des Projektes Mitgliederdatenbank für den StuRa (I06) 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 Projekt diese Bedarfe erfüllt, werden in dem Use-Case Model und den System-Wide Requirements 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 die Mitgliederdatenbank des StuRas der HTW Dresden, die vom Team I6 entwickelt wird. Das System wird es dem StuRa erlauben, eine Mitgliederdatenbank zu nutzen, um damit ein hauseigenes System zu besitzen, welches ihnen ermöglicht, sämtliche Mitglieder des StuRas zu verwalten.

1.1.3. Definitionen, Akronyme und Abkürzungen

Siehe [Glossar](#).

1.2. Positionierung

1.2.1. Fachliche Motivation

Der Studierendenrat der HTW Dresden beschäftigt sich mit den Wünschen, Problemen und Anforderungen der Studierenden der HTW. Er arbeitet eng mit der Hochschulleitung und des Studentenwerkes des Landes Sachsen zusammen um die Studienbedingungen zu verbessern. Hierfür fallen zahlreiche Arbeiten an, welche durch eine entsprechende Anzahl von Mitgliedern im StuRa bearbeitet werden müssen. Zur Zeit sind 205 Mitglieder (Stand 19.12.2022) im StuRa tätig. Hierfür existiert eine Excel-Tabelle, die die Mitglieder listet und zusätzlich Informationen über ihre Position im StuRa beinhaltet. Bei dieser Zahl an Angehörigen verliert man schnell den Überblick und eine ordentliche Organisation ist nahezu unmöglich.

Dieses Projekt beschäftigt sich mit den angesprochenen Problemen. Hierfür soll ein hauseigenes System entwickelt werden, welches es ermöglicht alle Mitglieder übersichtlich zu listen und zu bearbeiten, neue Mitglieder einfach anzulegen und ausgeschiedene zu entfernen. Außerdem sollen Arbeitsaufgaben im System verteilt werden können.

In der Vergangenheit wurde bereits eine Software erstellt, die allerdings nicht voll funktionsfähig

war. Hier sollen, auf Anraten des StuRa, Teile des Layouts bzw. des Codes der alten Software übernommen werden.

1.2.2. Problem Statement

Das Problem	Unübersichtliche und unpraktische Organisation der Mitglieder im StuRa
betrifft	den StuRa
die Auswirkung davon ist	Unübersichtlichkeit; Fehleranfälligkeit; unzureichende Bearbeitungsmöglichkeit
eine erfolgreiche Lösung wäre	ein System, welches Mitglieder übersichtlich listet und Bearbeitung zulässt, einfach hinzufügen und löschen lässt und ggf. zusätzliche Anwendungen

1.2.3. Positionierung des Produkts

Für	Mitglieder des StuRas
die	Pflege der Mitglieder zulässt
Das Produkt / die Lösung ist eine	hausteigene Software
die	die Daten für alle Mitglieder organisiert
Im Gegensatz zu	zu Excel Tabellen
Unser Produkt	ermöglicht die übersichtliche Organisation der Mitglieder des StuRas

1.3. Stakeholder Beschreibungen

1.3.1. Zusammenfassung der Stakeholder

Name	Beschreibung	Verantwortlichkeiten
HTW Dresden	Hochschule	stellt Studierende für StuRa, zahlt AE an die jeweiligen Studierenden
Mitglieder StuRa	Vertreter der Studierenden der HTW	Nutzer des Produkts
Systembetreuer, Admin	Systemadministrator	Für die Funktionalität, Wartung und Aktualisierung des Produkts verantwortlich
Hacker	Personen, die das System von außen stören können	Gefahr, da sensible Daten ausgetauscht werden

1.3.2. Benutzerumgebung

1. Anzahl der Personen, die an der Erfüllung der Aufgabe beteiligt sind ändert sich vorerst nicht.

2. Die Bearbeitungszeit soll schnell gehen. Hierfür werden je nach Datenumfang 0-2 Minuten geschätzt
3. Das Produkt soll an jedem Computer verwendbar und intuitiv bedienbar sein.
4. Als Systemplattformen werden Windows und Linux eingesetzt.

1.4. Produkt-/Lösungsüberblick

1.4.1. Bedarfe und Hauptfunktionen

Bedarf	Priorität	Features	Bemerkungen
einfache Verwaltung der Mitglieder	hoch	Eigener Button für "Mitglieder", in dem ein bestehendes Mitglied bearbeitet werden kann. Hier sollen auch neue Mitglieder angelegt bzw. gelöscht werden können.	bereits von Vorgänger implementiert
Ausgabe der Mitwirkungsdauer von Mitgliedern	mittel	Es soll ein eigener Button "Mitwirkungsdauer" entstehen. Hier können von allen oder nur bestimmten Mitgliedern die Mitwirkungsdauer eingesehen werden.	von Auftraggeber nicht mehr gewünscht
Ausgabe der Bearbeitungshistorie	niedrig	Eigener Button der es Admins ermöglicht, die Bearbeitungshistorie anzeigen zu lassen.	bereits von Vorgänger implementiert
Daten im- und exportieren	niedrig	Es muss möglich sein, vorhandene Daten in die Mitgliederdatenbank aufzunehmen und auch wieder in bestimmten Formaten auszugeben	Import von Vorgänger ermöglicht, Export nicht
Ausgabe von Stimmberechtigungen	mittel	Es sollte möglich sein, dass über einen eigenen Button alle stimmberechtigten StuRa-Mitglieder in einer Liste aufgelistet werden	von Auftraggeber nicht mehr gewünscht

1.5. Zusätzliche Produktanforderungen

Anforderung	Priorität
Einfache und intuitive Bedienbarkeit	hoch
System nur online nutzbar	hoch
System muss auf Windows, Linux und gängigen Browsern nutzbar sein	hoch
Unterteilung in Handlungsmöglichkeiten → Admin, untergeordneter User, Betrachter	mittel

2. Use-Case Model: Mitgliederdatenbank für den StuRa (I06)

2.1. Allgemeine Informationen

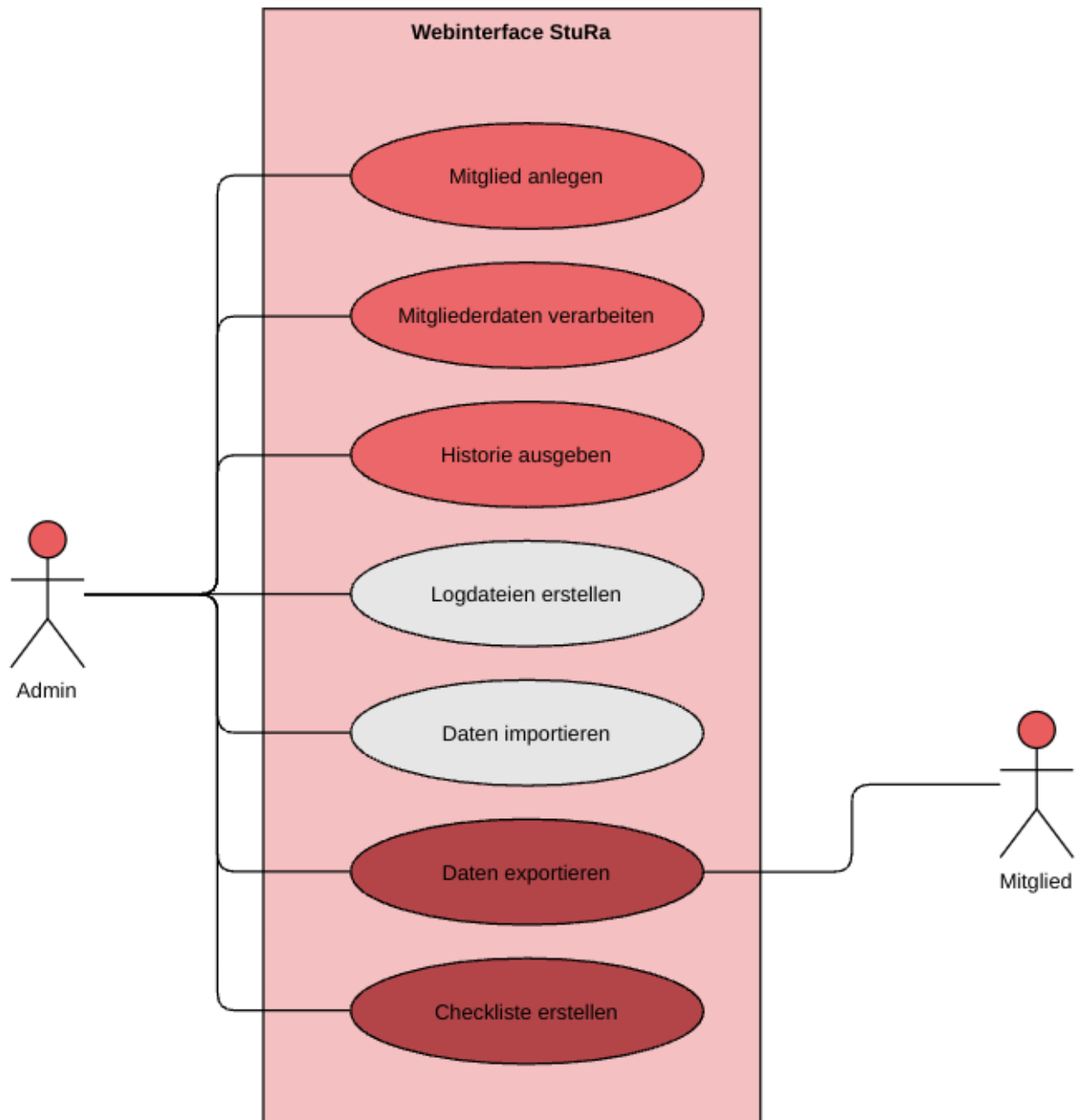
Im nachfolgenden Abschnitt werden alle identifizierten Use Cases, deren Akteure und deren Abläufe vorgestellt. Da ein bereits vorhandenes Projekt weiterentwickelt wird, wurden auch Use Cases identifiziert, die bereits implementiert wurden. Dementsprechend werden einige dieser bereits implementierten Use Cases weiterentwickelt sowie neue Use Cases hinzugefügt.

2.2. Identifizierte Use Cases

- **UC01:** Mitglied anlegen (bereits vorhanden - wird weiterentwickelt)
- **UC02:** Mitgliederdaten bearbeiten (bereits vorhanden - wird weiterentwickelt)
- **UC03:** Historie ausgeben (bereits vorhanden - wird weiterentwickelt)
- **UC04:** Logdateien erstellen (bereits vorhanden)
- **UC05:** Daten importieren (bereits vorhanden)
- **UC06:** Daten exportieren (neu)
- **UC07:** Checkliste erstellen (neu)

Bereits vorhandene Use Cases sind im Use Case Diagram grau hinterlegt, vorhandene Use Cases, welche weiterentwickelt wurden mit rot und neue Use Cases mit dunkelrot.

2.3. Use Case Diagramm



2.4. Use-Case: UC01 Mitglied anlegen

2.4.1. Kurzbeschreibung

Der Use Case beschreibt das Anlegen neuer Nutzer im Online-Tool. Diese sollen mit den Rechten, Aufgaben und Stundensätzen ausgestattet werden, die zu ihrem jeweiligen Amt passen.

2.4.2. Kurzbeschreibung der Akteure

Die StuRa-Mitglieder wollen ein neues Mitglied anlegen und dieses dem zugehörigen Amt zuordnen.

2.4.3. Vorbedingungen

- User muss im Online-Tool angemeldet sein.

2.4.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt mit einem Klick auf den "Mitgliederverwaltung"-Button.
2. Das System zeigt eine Übersicht aller, im Online-Tool registrierten, Mitglieder an.
3. Der User betätigt den Button "Neues Mitglied".
4. Das System gibt eine Auswahl an vorgefertigten Ämtern vor.
5. Der User wählt ein Amt aus.
6. Das System ordnet dem Mitglied Rechte, Aufgaben und Stundensatz entsprechend des Amtes zu.
7. Der User gibt die geforderten Daten für das neue Mitglied ein.
8. Das System überprüft die Eingaben auf ihre Vollständigkeit und Korrektheit.
9. Das System speichert das neue Mitglied ab.
10. Der Use Case endet.

2.4.5. Alternative Abläufe

Alternativer Ablauf 1

1. Der User hat nicht alle geforderten Daten angegeben.
2. Der Speichervorgang des neuen Mitglieds schlägt fehl.
3. Dem User wird eine entsprechende Fehlermeldung ausgegeben.

Alternativer Ablauf 2

1. Der User hat ungültige Daten angegeben.
2. Der Speichervorgang des neuen Mitglieds schlägt fehl.
3. Dem User wird eine entsprechende Fehlermeldung ausgegeben.

2.4.6. Nachbedingungen

- Alle eingegebenen Daten und ausgewählten Optionen wurden gespeichert.
- Neue Nutzer haben ihre korrekten Ämter, Funktionen und Rechte.

2.4.7. Besondere Anforderungen

- Nur User mit Adminrechten dürfen Mitglieder anlegen.

Usability

- Der Use Case gehört zu der grundlegenden Funktionalität des Systems und muss dementsprechend eine einfache Bedienbarkeit ermöglichen.

2.5. Use-Case: UC02 Mitgliederdaten bearbeiten

2.5.1. Kurzbeschreibung

Die vorhandenen Daten benötigen eine Änderung.

2.5.2. Kurzbeschreibung der Akteure

Die StuRa-Mitglieder wollen die Daten eines bereits vorhandenen Mitglieds händisch ändern.

2.5.3. Vorbedingungen

- User muss im Online-Tool angemeldet sein.
- Das zu ändernde Mitglied ist im System erfasst.

2.5.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt mit einem Klick auf den "Datenverwaltung"-Button
2. Das Tool zeigt die vorhandenen Mitgliedsdaten an.
3. Der User wählt ein Mitglied, das bearbeitet werden soll.
4. Eine Checkliste gibt Auskunft über noch offene Aufgaben, die der User bearbeiten muss.
5. Der User ändert fehlerhafte oder ergänzt fehlende Daten.
6. Der User bestätigt die Änderung der Daten.
7. Das System überprüft die Eingaben auf ihre Vollständigkeit und Korrektheit.
8. Das System speichert alle Daten in der Datenbank.
9. Der Use Case endet.

2.5.5. Alternative Abläufe

Alternativer Ablauf 1

1. Der User hat nicht alle geforderten Daten angegeben bzw. geforderte Daten gelöscht.
2. Der Speichervorgang der Änderungen schlägt fehl.
3. Dem User wird eine entsprechende Fehlermeldung ausgegeben.

Alternativer Ablauf 2

1. Der User hat ungültige Daten angegeben.
2. Der Speichervorgang der Änderungen schlägt fehl.
3. Dem User wird eine entsprechende Fehlermeldung ausgegeben.

2.5.6. Nachbedingungen

- Das System hat die geänderten Daten vollständig gespeichert.

2.5.7. Besondere Anforderungen

- Nur User mit Adminrechten dürfen Mitgliederdaten bearbeiten.

Usability

- Der Use Case gehört zu der grundlegenden Funktionalität des Systems und muss dementsprechend eine einfache Bedienbarkeit ermöglichen.

2.6. Use-Case: UC03 Historie ausgeben

2.6.1. Kurzbeschreibung

Es sollen alle historischen Änderungen der Daten angezeigt werden.

2.6.2. Kurzbeschreibung der Akteure

Die StuRa-Mitglieder wollen alle historischen Änderungen an den Daten einsehen.

2.6.3. Vorbedingungen

- User muss im Online-Tool angemeldet sein.

2.6.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt mit einem Klick auf den "Datenverwaltung"-Button
2. Das Tool zeigt die vorhandenen Mitgliedsdaten an.
3. Der User betätigt den "Historie"-Button.
4. Das System zeigt alle historischen Änderungen an der Datenbank an, sowie durch wen sie durchgeführt wurden.
5. Der Use Case endet.

2.6.5. Besondere Anforderungen

- Nur User mit Adminrechten dürfen sich die Historie ausgeben lassen.

2.7. Use-Case: UC04 Logdateien erstellen

2.7.1. Kurzbeschreibung

Es sollen die Protokolldateien ausgegeben werden, um einen Einblick in die Netzwerkereignisse zu erhalten.

2.7.2. Kurzbeschreibung der Akteure

Die StuRa-Mitglieder wollen Logdateien erstellen um sich einen Überblick über aufgetretene Ereignisse, einen Prozess-Status oder Transaktionen zu verschaffen.

2.7.3. Vorbedingungen

- User muss im Online-Tool angemeldet sein.

2.7.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt automatisch bei Benutzung des Systems.
2. Aktionen und Prozesse werden protokolliert.
3. Der Use Case endet.

2.8. Use-Case: UC05 Daten importieren

2.8.1. Kurzbeschreibung

Es sollen vorhandene Daten, beispielsweise aus einer Excel-Tabelle, in das Datenbank-Tool importiert werden.

2.8.2. Kurzbeschreibung der Akteure

Die StuRa-Mitglieder wollen vorhandene Daten in die Datenbank importieren lassen.

2.8.3. Vorbedingungen

- User muss im Online-Tool angemeldet sein.

2.8.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt mit einem Klick auf den "Datenverwaltung"-Button
2. Das Tool zeigt die vorhandenen Mitgliedsdaten an.
3. Der User wählt den Button "Datenimport".
4. Der User wählt eine Datei mit den benötigten Daten aus.
5. Die Daten werden importiert.
6. Der Use Case endet.

2.8.5. Alternative Abläufe

Alternativer Ablauf 1

1. User wählt eine Datei mit einem nicht unterstützten Dateiformat aus.
2. Das System gibt eine entsprechende Fehlermeldung aus.
3. Der Prozess wird abgebrochen.

2.8.6. Nachbedingungen

- Alle importierten Daten wurden korrekt gespeichert.

2.8.7. Besondere Anforderungen

- Nur User mit Adminrechten dürfen Daten importieren.

2.9. Use-Case: UC06 Daten exportieren

2.9.1. Kurzbeschreibung

Die vorhandenen Daten sollen zur Sicherung exportiert werden.

2.9.2. Kurzbeschreibung der Akteure

Die StuRa-Mitglieder wollen die in der Datenbank vorhandenen Daten exportieren.

2.9.3. Vorbedingungen

- User muss im Online-Tool angemeldet sein.

2.9.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt mit einem Klick auf den "Datenverwaltung"-Button
2. Das Tool zeigt die vorhandenen Mitgliedsdaten an.
3. Der User wählt den Button "Datenexport".
4. Der User wählt einen Ort zur Sicherung der Daten aus.
5. Die Daten werden exportiert.
6. Der Use Case endet.

2.10. Use-Case: UC07 Checkliste erstellen

2.10.1. Kurzbeschreibung

Es soll eine Checkliste erstellt werden, die ausgibt, welche Berechtigungen den Mitgliedern noch übergeben werden müssen (z.B. Schlüssel, Websitegruppen etc.).

2.10.2. Kurzbeschreibung der Akteure

Die StuRa-Mitglieder wollen sich über zu übergebenden Berechtigungen informieren.

2.10.3. Vorbedingungen

- User muss im Online-Tool angemeldet sein.

2.10.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt bei Neuanlage oder Änderung eines Mitglieds.
2. Es wird eine Checkliste generiert, die zeigt welche Berechtigungen dem Mitglied noch übergeben werden müssen bzw. welche bereits erledigt sind.
3. Der User kann sich die Checkliste ausgeben lassen mit einem Klick auf den "Checkliste"-Button.
4. Der Use Case endet.

2.11. Wireframes

Hierbei handelt es sich um Wireframes für unseren Prototypen aus Software Engineering 1. Im Zuge der Übernahme des Vorgängerprojektes sind diese überflüssig geworden.



Mitglied hinzufügen

Vorname*

Nachname*

s-Nummer*

E-Mail*

alternative E-Mail (optional)

Nickname (optional)

Website (optional)

Amt*



Abbildung 1. Formular



Mitglied hinzufügen

Vorname*

Max

Nachname*

Mustermann

s-Nummer*

s12345

E-Mail*

|

Please fill out this field.

nal)

Nickname (optional)

Website (optional)

Amt*

kooptiertes (beratendes) Mitglied



Mitglied hinzufügen

Abbildung 2. Input-Validierung

Mitglied hinzufügen

Vorname*

Max

Nachname*

Mustermann

s-Nummer*

s12345

E-Mail*

alternative E-Mail (optional)

Nickname (optional)

Website (optional)

Amt*

kooptiertes (beratendes) Mitglied



kooptiertes (beratendes) Mitglied

Leitung Bereich Rekrutierung

Leitung Referat Öffentlichkeitsarbeit

...

Abbildung 3. Formular (Ämter-Auswahl)

3. System-Wide Requirements: Mitgliederdatenbank für den StuRa (I06)

3.1. Einführung

In diesem Dokument werden die systemweiten Anforderungen für das Projekt Mitgliederdatenbank 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



Die funktionalen Anforderungen, die sich aus der Interaktion von Nutzern mit dem System ergeben, sind als Use Cases in einem separaten Dokument festgehalten. [hier ggf. Verweis einfügen]

3.2. Systemweite funktionale Anforderungen

- **SWFA-1: Ein Access-Control für das Backend/Verwaltungsoberfläche ist nötig.**
 - Zur Überprüfung können mittels unberechtigter oder gar falscher Logins Zugriffe auf unberechtigte Daten oder Use Cases versucht werden.
- **SWFA-2: Änderungen am System oder an der Software sollen dokumentiert werden (Auditing) und leicht nachvollziehbar sein.**
 - Zur Validierung können mit verschiedenen Test-Nutzern Daten verändert, gelöscht und hinzugefügt werden. Anschließend werden alle getätigten Änderungen mit der Historie abgeglichen.
- **SWFA-3: Das System muss relevante Daten persistent speichern.**
 - Zur Sicherstellung kann eine Reihe plötzlicher Systemausfälle simuliert werden. Bei einer anschließenden Überprüfung werden relevante Daten auf Vollständigkeit überprüft.

3.3. Qualitätsanforderungen für das Gesamtsystem

3.3.1. Benutzbarkeit (Usability)

- **NFAU-1 Lokalisation:**

Die Stakeholder sitzen ausschließlich in Deutschland, weshalb die Software deutsch lokalisiert sein soll (Sprache, Datumsformat, Währung).
- **NFAU-2 Erlernbarkeit:**

Es soll sichergestellt sein, dass auch ein ungelernter Nutzer mithilfe einer Anleitung den Umgang mit der Software problemlos und in absehbarer Zeit (max. 1 Tag) erlernen kann.

- **NFAU-3 Zugangsbeschränkung:**

Mithilfe einer Rechteverwaltung soll sichergestellt werden, dass nur berechtigte Nutzer Schreibrechte haben. Die Berechtigungen sollen vom Administrator festgelegt werden können.

- **NFAU-4 Benutzerfreundlichkeit:**

Mithilfe von maximal 3 Klicks sollen relevante Informationen für Mitglieder in einem geordneten Format ausgegeben werden können. Die Ausgabe muss vom Benutzer nach mehreren Kriterien filterbar sein.

- **NFAU-5 Eingabehilfe:**

Das Tool soll zwischen verpflichtenden und optionalen Eingaben unterscheiden (z.B. Eingabe einer alternativen E-Mail-Adresse). Es ist die Aufgabe des Frontends, dem Nutzer diese Unterscheidung zu verdeutlichen.

3.3.2. Zuverlässigkeit (Reliability)

- **NFAR-1 Verfügbarkeit:**

Website sowie Datenbank benötigen keine 99,99% Erreichbarkeit, sollten jedoch so oft wie möglich verfügbar sein.

- **NFAR-2 Eingabevalidierung:**

Die Benutzereingaben sind auf Sinnhaftigkeit zu überprüfen um Falscheingaben zu vermeiden (z.B. keine Postleitzahl im Namensfeld). Fehlende Pflichtdaten führen zu einer Eingabeaufforderung an den Nutzer und lösen noch keine Änderung an der Datenbank aus.

- **NFAR-3 Dokumentation:**

Änderungen sollen dokumentiert werden (z.B. Nutzer A löscht Mitglied B am 01.01.2023). Systemausfälle sollen mittels Log-Dateien nachvollziehbar sein.

- **NFAR-4 Wiederherstellbarkeit:**

Im Zweifelsfall müssen Backups vom Nutzer einspielbar sein.

3.3.3. Effizienz (Performance)

- **NFAP-1 Serialisierung:**

Auf die Website sollen bis zu 3 Nutzer gleichzeitig zugreifen können. Datenbankabfragen sollten auch bei hoher Mitgliederanzahl innerhalb von 5 Sekunden ausgeführt sein.

- **NFAP-2 Workflow:**

Das Anlegen neuer Datensätze soll direkt erfolgen, um reibungslose Arbeitsabläufe zu gewährleisten.

3.3.4. Wartbarkeit (Supportability)

- **NFAS-1 Installation:**

Es soll eine Installationsanleitung beigelegt werden, welche das selbständige (neu-)installieren der Software durch die Nutzer ermöglicht.

- **NFAS-2 Erweiterbarkeit:**

Berechtigte Nutzer sollen Tabellen selbständig verwalten (z.B. Anlegen neuer Spalten) können.

3.4. Zusätzliche Anforderungen

3.4.1. Einschränkungen (Constraints)

- **ZAC-1 Plattform:**

Die Website sowie die Datenbank sollen auf einem Server des StuRa laufen. Des Weiteren soll die Website auf allen aktuell gängigen Browsern nutzbar sein.

- **ZAC-2 Ressourcen:**

Der StuRa stellt einen Docker zur Verfügung auf dem ein beliebiges Linux-Betriebssystem virtualisiert werden kann.

- **ZAC-3 Implementierung:**

Es soll kein Java verwendet werden. Das System soll unter Linux lauffähig sein.

3.4.2. Organisatorische Randbedingungen

- **ZAO-1 Standards:**

- Die Software muss dem Wertekanon des StuRa entsprechen
- Die Datensicherung muss ausschließlich lokal erfolgen

3.4.3. Rechtliche Anforderungen

- **ZAR-1 Lizenzierung:**

Die Anwendung soll Open-Source sein

- **ZAR-2 Datenschutz:**

Das System muss den Datenschutzerfordernungen der HTW Dresden, des StuRas und der DSGVO entsprechen

4. Glossar: Mitgliederdatenbank für den StuRa (I06)

4.1. Einführung

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

4.2. Begriffe

Begriff	Definition und Erläuterung	Synonyme
Access-Control	Reguliert wer oder was auf eine bestimmte Ressource zugreifen kann. Dies geschieht mittels Autorisierung und Authentifizierung.	Zugriffskontrolle
Backend	Bezeichnet den Teil eines IT-Systems, welcher sich mit der Datenverarbeitung im Hintergrund beschäftigt. Die Daten erhält es vom Frontend.	
Frontend	Die tatsächliche Schnittstelle zwischen Nutzer und Software-Anwendung. In der Regel geschieht dies über eine grafische Benutzeroberfläche.	Präsentationsebene
Administrator	Eine Person mit Vollzugriff auf das System. Der Admin ist Zuständig für die Wartung und Installation eines IT-Systems.	Admin
Online-Tool	(Hilfs-)Werkzeug zur Datenverarbeitung mithilfe einer Benutzeroberfläche.	Online-Plattform
Auditing	Feststellung der Datensicherheit sowie Datenintegrität.	
persistente Speicherung	Speicherung von Daten auch über die Laufzeit des Systems hinaus.	nicht flüchtige Datenspeicherung
Beratendes Mitglied	In den StuRa gewähltes Mitglied.	

Begriff	Definition und Erläuterung	Synonyme
Exekutives Mitglied	Mitglied mit einer ausführenden Funktion	
Legislatives Mitglied	Von den Studierenden gewähltes Mitglied	
Legislatur	Wahlperiode eines gewählten Mitgliedes	
Plenum	Zusammenschluss sämtlicher Mitglieder des StuRas	

4.3. Abkürzungen und Akronyme

Abkürzung	Bedeutung	Erläuterung
StuRa	Studentenrat	
UP	Unified Process	Vorgehensmodell für die Softwareentwicklung

4.4. Verzeichnis der Datenstrukturen

Bezeichnung	Definition	Format	Gültigkeitsregeln	Aliase
Anmeldedaten	Zusammensetzung von Benutzername und Passwort.	String	Emailadresse muss @-Zeichen und . Punkt enthalten.	Login
Mitgliedsdaten	Gesamtheit personenbezogener Daten eines Mitglieds (Name, Adresse, Amt)	Klasse		-

5. Domain Model: Mitgliederdatenbank für den StuRa (I06)

