

Entwurfsdokumentation

Mitgliederdatenbank für den StuRa (I06)

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

26. Juni 2023

Inhaltsverzeichnis

1. Architecture Notebook: Mitgliederdatenbank für den StuRa (I06)	1
1.1. Zweck	1
1.2. Architekturziele und Philosophie	1
1.3. Annahmen und Abhängigkeiten	1
1.4. Architektur - relevante Anforderungen	1
1.5. Entscheidungen, Nebenbedingungen und Begründungen	2
1.6. Architekturmechanismen	2
1.7. Wesentliche Abstraktionen	3
1.8. Schichten oder Architektur-Framework	3
1.9. Architektursichten (Views)	3

1. Architecture Notebook: Mitgliederdatenbank für den StuRa (I06)

1.1. Zweck

Dieses Dokument beschreibt die Philosophie, Entscheidungen, Nebenbedingungen, Begründungen, wesentliche Elemente und andere übergreifende Aspekte des Systems, die Einfluss auf Entwurf und Implementierung haben.

1.2. Architekturziele und Philosophie

Das System stellt ein Online-Tool zur Verwaltung der Mitglieder des StuRa der HTW Dresden dar. Offizielle Mitglieder des StuRa erhalten über einen Login Zugang zur Website und können entsprechend ihrer verliehenen Rechte entweder nur Informationen einsehen oder diese gegebenenfalls auch bearbeiten. Auf Grund dessen muss eine parallele Nutzung von bis zu drei Personen gewährleistet sein.

Eine gute Bedienbarkeit wird durch eine übersichtliche, leicht verständliche und intuitiv bedienbare Benutzeroberfläche sichergestellt. Die leichte Bedienbarkeit soll ebenfalls garantieren, dass auch ungelernte Nutzer mithilfe einer Anleitung den Umgang mit der Software problemlos und in absehbarer Zeit erlernen können.

Des Weiteren sollen schnelle Seitenladezeiten geboten werden und Datenbankabfragen sollen auch bei hoher Mitgliederanzahl innerhalb von 5 Sekunden ausgeführt werden.

1.3. Annahmen und Abhängigkeiten

Annahmen

- Jeder Nutzer besitzt eine stabile Internetverbindung mit genügend Bandbreite
- Jeder Nutzer nutzt einen aktuellen Browser (z.B. Firefox oder Chrome) und erlaubt Javascript
- Der Server auf dem die Webseite laufen soll bietet ausreichend Ressourcen
- Die bisher verwendeten Datenbankmodelle und Frameworks können weiter verwendet werden
- Die Mitgliederdatenbank wird in Zukunft funktional erweitert
- Das Django-Framework gibt bereits eine Aufbau-Architektur vor

Abhängigkeiten

- Wir sind von den gegebenen Ressourcen des Sturas abhängig. Diese belaufen sich auf einen Linux Docker-Container des hauseigenen Servers

1.4. Architektur - relevante Anforderungen

- SWFA-1 Access-Control

- NFAU-3 Zugangsbeschränkung
- NFAP-1 Serialisierung
- ZAC-1 Plattform

1.5. Entscheidungen, Nebenbedingungen und Begründungen

- Bei der Wahl der Programmiersprache haben wir uns für Python entschieden, da uns durch den StuRa vorgegeben wurde kein Java zu verwenden. Des Weiteren bietet Python eine gut verständliche Syntax und ist vielfältig einsetzbar.
- Wir entschieden uns für das Django Webframework, da dieses eine große Community und somit großen Support bietet. Zusätzlich ist es ein Open-Source-Projekt.
- Wir haben uns für ein Webdesign entschieden, welches der StuRa-Website ähnelt, um eine vertraute Nutzerumgebung zu schaffen.
- Für das Datenbanksystem werden wir SQLite verwenden, da dies bereits in den Vorjahren genutzt wurde. Außerdem ist es Open-Source und wird standardmäßig von Django unterstützt.

1.6. Architekturmechanismen

[Doku "Concept: Architectural Mechanism"](#)

Archivierung

- Zweck: Es dürfen auch bei Systemausfällen keine Datenverluste entstehen
- Eigenschaften: Daten werden regelmäßig gesichert und sind nach Systemfehlern oder -ausfall wiederherstellbar
- Funktion: Backups werden vom Administrator manuell angelegt und müssen lediglich einfach einspielbar sein.

Kommunikation

- Zweck: Datenaustausch zwischen Website und Datenbank
- Eigenschaften: synchron, schnell
- Funktion: Mittels optimierter SQL-Anfragen wird eine reibungslose Kommunikation gewährleistet

Persistenz

- Zweck: Daten müssen dauerhaft für den Admin und die Mitglieder des Stura verfügbar bleiben
- Eigenschaften: Speicherung der Daten in einer Datenbank
- Funktion: Die Datensätze werden noch während der Ausführung der Anwendung in einer Datenbank gespeichert und sind von dort auch wieder abrufbar

Zugriffsschutz

- Zweck: Schutz der personenbezogenen Daten vor unberechtigt Zugriff
- Eigenschaften:
 - Nicht jeder Nutzer ist dazu berechtigt, alle Daten einzusehen
 - Bestimmte Nutzer haben bestimmte Rechte, um verschiedene Informationen einsehen zu können
- Funktion: Anmeldung der Nutzer mit Logindaten, anhand derer ermittelt werden kann, welche Rechte der Nutzer hat. (Unterscheidung nach Mitglied Stura und Admin)

Error Management

- Zweck: Anwender muss über unerwartete Fehler informiert werden
- Eigenschaften: Fehlerausgabe im Browser
- Funktion: Datenbank- oder Webservice-Fehler werden dem Anwender über verständliche Fehlermeldungen mitgeteilt

Eingabelogik

- Zweck: Eingabemasken sollten mit geringem Aufwand ausfüllbar sein
- Eigenschaften: Leichte Bedienung durch Menüführung statt Texteingaben
- Funktion: Mittels Dropdown-Menü werden dem Anwender verschiedene Eingabemöglichkeiten vorgegeben, aus denen er mittels Mausklick auswählen kann

1.7. Wesentliche Abstraktionen

- Schutz vor Änderungen unbefugter Dritter
- Datensicherung durch relationale Datenbank
- robuste Verwaltung von Mitgliedern und Ämtern
- Ausgabe einer Checkliste/Arbeitsleitfaden für den StuRa

1.8. Schichten oder Architektur-Framework

Die Webanwendung basiert wie in den Vorjahren auf dem Webframework Django. Es handelt sich um eine Komponentenarchitektur. Die Objekte der einzelnen Klassen werden voraussichtlich in einer Datenbank mittels SQLite gespeichert. Auf einem dafür vorgesehenen Linux Docker-Container des StuRa-Webservers soll die Webanwendung gehostet werden.

1.9. Architektursichten (Views)

Folgende Sichten werden empfohlen:

1.9.1. Logische Sicht

Paketdiagramm (mit Klassen)

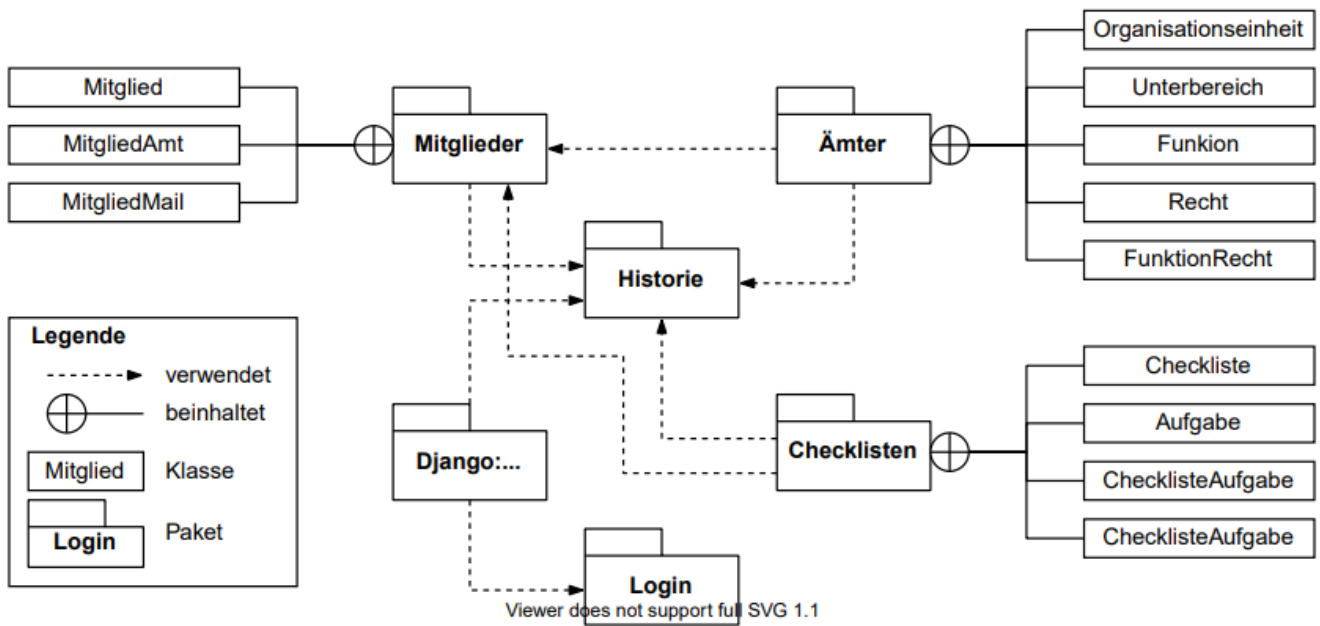


Abbildung 1. (Von Vorgängergruppe übernommen, Klick auf Bild führt zum Original)

1.9.2. Physische Sicht (Betriebssicht)

- Webserver → LXC Container → Webanwendung (Django)

1.9.3. Use cases

- UC01 Mitglied anlegen
- UC02 Mitgliederdaten bearbeiten
- UC03 Historie ausgeben
- UC05 Daten importieren
- UC06 Daten exportieren
- UC07 Checkliste erstellen