

SE I - Belegabgabe

Turnierauswertungssoftware

Julian Groß, Alexander Zajcev, Kareen Khouri, Jonas Griefhaber, Maximilian
Lohr, Tim Reimann, Benedikt Mühlhans

01. Januar 2025

Inhaltsverzeichnis

Projektdokumentation	1
1. Projektplan: Turnierauswertungssoftware	2
1.1. Einführung	2
1.2. Projektorganisation	2
1.3. Praktiken und Bewertung	3
1.4. Meilensteine und Ziele	3
1.5. Deployment	3
1.6. Erkenntnisse (Lessons learned)	3
Technische Spezifikation	5
2. Vision: Turnierauswertungssoftware	6
2.1. Einführung	6
2.2. Positionierung	7
2.3. Stakeholder Beschreibungen	7
2.4. Produkt-/Lösungsüberblick	8
3. Glossar: Turnierauswertungssoftware	10
3.1. Einführung	10
3.2. Begriffe	10
3.3. Abkürzungen und Akronyme	11
3.4. Verzeichnis der Datenstrukturen	11
4. Domain Model: Turnierauswertungssoftware	12
4.1. Allgemeine Informationen	12
4.2. Domain-Model-Diagramm	12
5. UX-Konzept: Turnierauswertungssoftware	13
5.1. Einführung	13
5.2. Anforderungen und Prioritäten	13
5.3. Gestaltung der Benutzeroberfläche	13
Entwurfsdokumentation	15
6. Architecture Notebook: Turnierauswertungssoftware	16
6.1. Links zur Erläuterung	16
6.2. Zweck	16
6.3. Architekturziele und Philosophie	16
6.4. Annahmen und Abhängigkeiten	16
6.5. Architektur-relevante Anforderungen	16
6.6. Entscheidungen, Nebenbedingungen und Begründungen	17
6.7. Architekturmechanismen	18
6.8. Wesentliche Abstraktionen	18
6.9. Schichten oder Architektur-Framework	19
6.10. Architektursichten (Views)	19

7. Test Cases: Turnierauswertungssoftware	21
7.1. Allgemeine Informationen	21
7.2. Test Cases	21

Projektdokumentation

- Projektplan

1. Projektplan: Turnierauswertungsoftware

1.1. Einführung

Dieser Projektplan beschreibt die Struktur, Methodik und Zielsetzungen des Teams und dient als lebendiges Dokument, das während der Projektumsetzung fortlaufend ergänzt und optimiert wird. Er unterstützt die Transparenz sowohl im Team als auch gegenüber externen Partnern, fördert den Informationsaustausch und hilft, die langfristigen Zielvorgaben des Projekts nicht aus den Augen zu verlieren.

1.2. Projektorganisation

Das Projektteam für die Entwicklung der Turnierauswertungsoftware der STURA der HTW besteht aus engagierten Mitgliedern mit klar definierten Rollen, die ihre individuellen Stärken in das Projekt einbringen. Ziel des Projekts ist es, eine benutzerfreundliche und zuverlässige Software zu entwickeln, die die Erstellung von Spielplänen, die Erfassung von Ergebnissen und die Auswertung von Turnieren effizient unterstützt.

Die Leitung des Projekts übernimmt Tim als Projektleiter. Seine Hauptaufgaben umfassen die Koordination des Teams, die Überwachung des Projektfortschritts sowie die Kommunikation mit den Stakeholdern.

Der Bereich Analyse wird von Benedikt und Kareen verantwortet. Ihre Aufgabe ist es, die Anforderungen an die Software zu erheben, zu dokumentieren und mit den Bedürfnissen der STURA-Mitglieder abzugleichen.

Der Bereich Entwurf liegt in den Händen von Julian, der die technische und funktionale Architektur der Software entwickelt.

Die Entwicklung wird von Alex und Max umgesetzt. Sie programmieren die Kernfunktionen der Software, kümmern sich um die Integration der verschiedenen Module und sorgen dafür, dass die erstellten Prototypen einsatzbereit sind.

Im Bereich Test ist Jonas verantwortlich. Er stellt sicher, dass die entwickelte Software fehlerfrei funktioniert. Seine Aufgaben umfassen die Erstellung von Testplänen, die Durchführung umfangreicher Funktionstests sowie die Überprüfung, ob die Software die definierten Anforderungen erfüllt.

Zusätzlich besteht eine enge Zusammenarbeit mit der STURA der HTW, insbesondere mit den Verantwortlichen der Sport-Abteilung. Die Kommunikation erfolgt über regelmäßige Meetings und einen gemeinsamen Austausch per Mail.

Intern kommuniziert das Projektteam über wöchentliche Meetings, die hauptsächlich zur Abstimmung und zur Klärung von Aufgaben dienen. Für die fortlaufende Zusammenarbeit nutzt das Team Tools wie GitHub für den schnellen Austausch sowie zur Aufgabenverfolgung. Visual Studio Code wurde genutzt, um Änderungen an den Dokumenten vorzunehmen und den Code für die Prototypen zu schreiben.

Durch diese klare Rollenverteilung und die effiziente Zusammenarbeit ist das Team optimal aufgestellt, um eine hochwertige und funktionale Turnierauswertungssoftware zu entwickeln, die den Anforderungen der STURA der HTW gerecht wird.

1.3. Praktiken und Bewertung

Im Rahmen des Projekts zur Entwicklung der Turnierauswertungssoftware für die STURA der HTW werden verschiedene Management- und technische Praktiken eingesetzt, um eine strukturierte und qualitativ hochwertige Umsetzung sicherzustellen. Dazu gehört eine iterative Entwicklung.

Die iterative Entwicklung bildet das zentrale Element des Projektansatzes. Das Team arbeitet in Iterationen von jeweils drei Wochen, in denen die Phasen Anforderungsanalyse, Entwurf, Implementierung und Test abgeschlossen werden. Jede Iteration beinhaltet eine Präsentation der Fortschritte, um Feedback von den Stakeholdern einzuholen. Damit wurde sichergestellt, dass das Projekt auf Kurs bleibt. Durch regelmäßige Retrospektiven wird der Entwicklungsprozess kontinuierlich verbessert, indem Stärken identifiziert und Schwächen adressiert werden.

1.4. Meilensteine und Ziele

Iteration	Primary objectives (risks and use case scenarios)	Scheduled start or milestone	Target velocity
I1	<ul style="list-style-type: none">• kennenlernen im Team• erste Bedarfsanalyse	24.10.2024/ 11.11.2024	
I2	<ul style="list-style-type: none">• Tiefgründige Bedarfsanalyse	12.11.2024/ 29.11.2024	
I3	* entwicklung erster Modelle * 1. Prototyp gestartet	30.11.2024/ 20.12.2024	
I4	<ul style="list-style-type: none">• Verfeinerung des 1. Prototyps• Anpassungen der bisherigen Ergebnisse durch neue Bedürfnisse	21.12.2024/ 24.01.2025	

1.5. Deployment

Die Software wird lokal auf einem Rechner des Stura per Docker eingerichtet und kann auch von dort gestartet werden. Bei Turnieren soll in der Turnhalle ein mobiler Hotspot auf dem Rechner des Stura eröffnet werden, womit sich die Teilnehmenden dann verbinden können. Weitere Details zur Bereitstellung bzw. Übergabe der Software an den Stura werden gegen Ende des nächsten Semesters hinzugefügt.

1.6. Erkenntnisse (Lessons learned)

- Die interne Absprache bei geteilten Aufgaben ist essenziell und darf unter keinen Umständen vernachlässigt werden.
- Die präzise Vorbereitung der Meetings gestaltet diese deutlich effektiver

- Die Kommunikation mit dem Auftraggeber ist der entscheidendste Faktor in der Bedarfsanalyse

Technische Spezifikation

- Vision
- Glossar
- Domänenmodel
- UX-Konzept

2. Vision: Turnierauswertungssoftware

2.1. Einführung

Dieses Dokument soll die wesentlichen Bedarfe und Funktionalitäten des Systems Turnierauswertungssoftware sammeln, analysieren und definieren. Der Fokus liegt auf den Fähigkeiten, die von Stakeholdern und adressierten Nutzern benötigt werden, und der Begründung dieser Bedarfe. Die Details, wie das System Turnierauswertungssoftware diese Bedarfe erfüllt, werden in der Use-Case und Supplementary Specification beschrieben.

2.1.1. Zweck

Die Vision dieses Projektes ist es, die wesentlichen Anforderungen an das System aus Sicht und mit den Begriffen der künftigen Anwender zu beschreiben. Das System soll die Turnierorganisation für alle Teilnehmer erleichtern und optimieren.

2.1.2. Gültigkeitsbereich (Scope)

Dieses Visions-Dokument bezieht sich auf das System Turnierauswertungssoftware, das von unserem Team entwickelt wird. Das System wird es den Nutzern erlauben, Turniere online/offline zu erstellen, zu verwalten und auszuwerten, um damit die Organisation und Auswertung von Turnieren wesentlich leichter zu gestalten.

Gehört dazu:

- Webbasierte Anwendung
- Turnierplan Erstellung
- Erfassung und Auswertung der Ergebnisse
- pdf Export ermöglichen
- Ergebnisse als Exceldatei speichern

Gehört nicht dazu:

- Entwicklung einer mobilen App
- Teameinteilung
- Erstellung der Teams
- Druckfunktion

2.1.3. Definitionen, Akronyme und Abkürzungen

Siehe [Glossar](#).

2.1.4. Referenzen

<https://miro.com/welcomeonboard/>

2.2. Positionierung

2.2.1. Fachliche Motivation

Die bisherige manuelle Turnierplanung ergibt einen hohen Aufwand bei der Erstellung des Turnierplans und der Erfassung der Turnierergebnisse. Eine softwaregestützte Lösung spart Zeit bei der Organisation und bietet den Spielenden mehr Eigenverantwortung, da sie ihre Ergebnisse selbstständig eintragen können.

2.2.2. Problem Statement

Das Problem	Die Organisation und Auswertung des Volleyballturniers ist zeitaufwendig.
betrifft	Turnierorganisatoren und Teilnehmer.
die Auswirkung davon ist	Hoher Aufwand für die Organisatoren des Turniers.
eine erfolgreiche Lösung wäre	Eine Entlastung für die Organisatoren und mehr Transparenz für die Teilnehmer, da sie zu jeder Zeit digital den Spielplan und die Ergebnisse abrufen können. Zusätzlich können die Teilnehmer selbstständig Ergebnisse eintragen.

2.2.3. Positionierung des Produkts

Für	Organisatoren und Teilnehmer des Volleyballturniers
die	mehr Effizienz, Transparenz und Eigenverantwortung bei der Verwaltung des Turniers benötigen.
Das Produkt /	ist eine Softwarelösung für Turnierplanung und Verwaltung.
Die /	automatisierte Spielpläne und Verwaltung der Ergebnisse bietet.
Im Gegensatz zu	manueller Planung auf Papier.
Unser Produkt	verringert Aufwand für die Organisatoren und bezieht Teilnehmer bei der Verwaltung der Ergebnisse mit ein.

2.3. Stakeholder Beschreibungen

2.3.1. Zusammenfassung der Stakeholder

Name	Beschreibung	Verantwortlichkeiten
Organisatoren.	Verantwortlich für die Organisation des Turniers.	Einordnung der Teams, Erstellung des Spielplans, Verwaltung der Ergebnisse.

Name	Beschreibung	Verantwortlichkeiten
Turnierteilnehmer.	Volleyballspieler.	Führen die Partien aus und tragen Ergebnisse in den Turnierplaner ein.
Gesetzgeber.	Stellt gesetzliche Anforderungen sicher, die von der Software eingehalten werden müssen.	Vorgaben zur Einhaltung von Datenschutz im Bezug auf persönliche Daten.
Entwickler.	(Weiter-)entwicklung der Software.	Zuständig für die Funktionalität.

2.3.2. Benutzerumgebung

- Turnierorganisation: Eine Person (Lotte Richter).
- Spieler: 20 bis zu maximal 200 Personen.
- Planung des Turniers erfordert mehrere Stunden im Vorfeld.
- Manuelle Organisation und Auswertung der Turnierergebnisse vor Ort auf Papier.
- Volleyballturnier findet in einer Turnhalle mit schlechter Internetverbindung statt.

2.4. Produkt-/Lösungsüberblick

2.4.1. Bedarfe und Hauptfunktionen

Bedarf	Priorität	Features	Geplantes Release
Automatisierte Spielplanerstellung und -anzeige	Hoch	Erstellung und Export eines Spielplans	Version 1.0
Erfassung der Turnierergebnisse	Hoch	Speichern der Ergebnisse -lokal wenn kein Netz verfügbar ist- und Synchronisierung bei Netzverfügbarkeit	Version 1.1
Turnierauswertung	Hoch	Gewinnerteam soll basierend auf den Ergebnissen ermittelt und angezeigt werden	Version 1.1
Speicherung und nachträgliche Bearbeitung	Mittel	Ergebnisse sollen auch nach dem Turnier aufgerufen werden und verändert werden können	Version 1.2
Wiederverwendbarkeit	Mittel	Für jedes zukünftige Jahr soll ein neues Turnier erstellt werden können	Version 1.3

2.4.2. Zusätzliche Produktanforderungen

Anforderung	Priorität	Geplantes Release
Kompatibilität mit Mobilgeräten	Hoch	Version 1.3
PDF-Export des Turnierplans	Hoch	Version 1.0
Fehlerprüfung bei Ergebnissen	Mittel	Version1.2
Export der Ergebnisse als Excel-Datei	Mittel	Version1.2
Ein Turnier, welches gespeichert wurde, muss überschrieben werden können. (Möglicherweise Sicherheitsfrage vorschalten).	Mittel	Version 1.4

3. Glossar: Turnierauswertungssoftware

3.1. Einführung

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

3.2. Begriffe

Begriff	Definition und Erläuterung	Synonyme
Leistungsgruppe	Zusammenfassung von Teams mit ähnlichem spielerischen Niveau	keine
Team	Personengruppe, die zusammen gegen Andere spielt	Mannschaft
Spiel	Ein Aufeinandertreffen, wo zwei Teams gegeneinander spielen	Match, Partie, Satz, Runde
Spielplan	Plan, der die Reihenfolge der Spiele der teilnehmenden Teams regelt	Spiel-Übersicht, Turnierplan
Spielstand	Anzahl der nach Ende eines Spiels erzielten Punkte pro Mannschaft	Score, Punktestand
Feld	Bereich, auf dem ein Spiel stattfindet	Spielfläche
Schiedsrichter	Person, die ein Spiel leitet	Unparteiischer, Kampfrichter, Referee
Spieler	Person, die als Teil eines Teams am Turnier teilnimmt	Teilnehmer, Player, Sportler, Volleyballer
Turnier	Veranstaltung, bei der die Teams aufeinandertreffen	Wettkampf, Wettbewerb
Aussetzen	Zeit, in der ein Team weder spielt noch den Schiedsrichter-Job übernimmt	Pause
Software	Die Turnierauswertungssoftware die entwickelt werden soll	Turnierauswertungssoftware, Programm, Application
Organisatoren	Die Person, die das Turnier leiten.	Turnierleitung

Begriff	Definition und Erläuterung	Synonyme
Nutzer	All diejenigen, die die Software beim Turnier benutzen (v.a. Spieler und Turnierleitung).	User, Benutzer

3.3. Abkürzungen und Akronyme

Abkürzung	Bedeutung	Erläuterung
z.B.: A1	Leistungsgruppe A, Teamnummer 1	im Bezug auf Teams
Schiri	Schiedsrichter	s. Begriffe
Stura	Studentenrat	Auftraggeber und Veranstalter des Turniers
LG	Leistungsgruppe	Zusammenfassung von Teams mit ähnlichem spielerischen Niveau

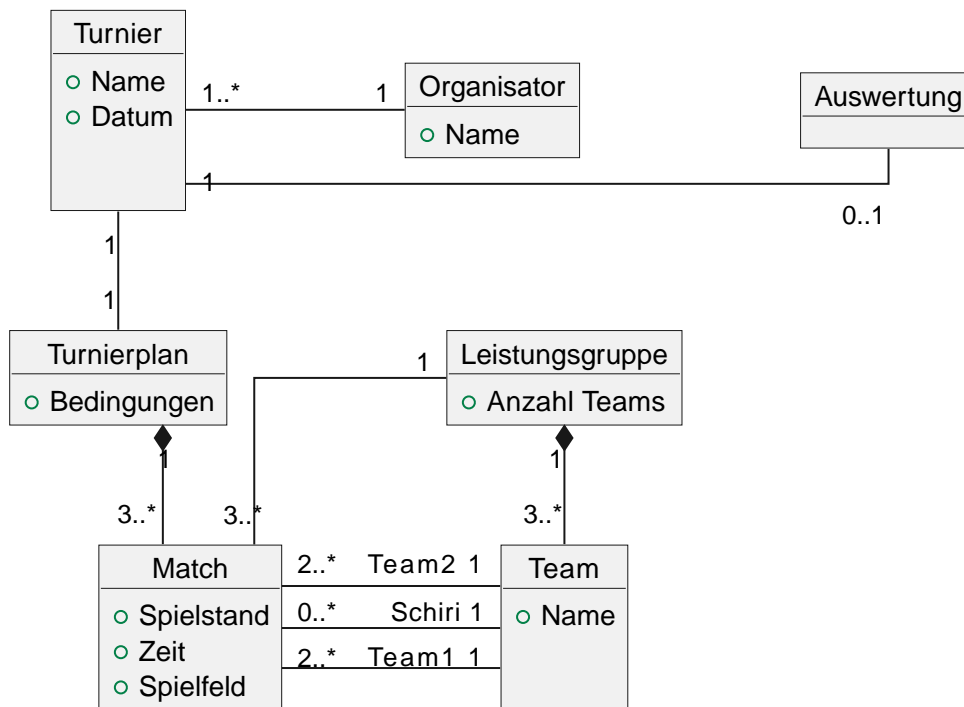
3.4. Verzeichnis der Datenstrukturen

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

4. Domain Model: Turnierauswertungsoftware

4.1. Allgemeine Informationen

4.2. Domain-Model-Diagramm



5. UX-Konzept: Turnierauswertungssoftware

5.1. Einführung

In diesem Dokument werden die Benutzeroberflächen des Systems Turnierauswertungssoftware beschrieben. Dies können sein:

- Wireframes relevanter Benutzeroberflächen
- Zusammenhänge mittels Taskflow-Diagrammen
- Angaben über die Gestaltung der Benutzeroberfläche, z.B. Styling, Bedienbarkeit, ...

5.2. Anforderungen und Prioritäten

Anforderung	Priorität	Geplantes Release
Teamnamen sollten vor Ort zu Beginn geändert werden können. Bsp: S1 = "Füchse"	Hoch	Version 1.4
Farben des Stura sollen verwendet werden	Hoch	Version 1.4
Bedienbarkeit soll so einfach wie möglich gehalten werden	Hoch	Version 1.4
Export der Ergebnisse als Excel-Datei	Mittel	Version 1.2
Spieler sollen sich über ihren Teamnamen am Anfang anmelden und bekommen dann ihren eigenen Turnierbaum angezeigt	Mittel	Version 1.4
Aktuelle Partie soll farblich (grün) hervorgehoben werden. Das soll erfolgen, nachdem das Ergebnis der vorherigen Partie eingetragen wurde	Mittel	Version 1.4
Legende einfügen, welche anzeigt, welches Team spielt und welches Team Schiedsrichter ist	Mittel	Version 1.4
Ein Turnier, welches gespeichert wurde, muss überschrieben werden können (möglicherweise Sicherheitsfrage vorschalten).	Mittel	Version 1.4
Gespielte Partien sollen auf dem Turnierplan durchgestrichen werden	Niedrig	Version 1.4
Die letzten 5 Turniere sollen in der Historie ersichtlich sein	Niedrig	Version 1.4

5.3. Gestaltung der Benutzeroberfläche

5.3.1. Farbgestaltung

Die Benutzeroberfläche wird in den Farben des Stura gestaltet. Diese Farben sind:

Primärfarbe: Orange

Sekundärfarbe: Schwarz, Grün

5.3.2. Wireframes

Startbildschirm:

Anzeige eines "Begrüßungsbildschirms" mit der Möglichkeit, sich über den Teamnamen anzumelden.

Schaltflächen: "Neues Turnier erstellen", "Turnier laden", "Historie anzeigen".

Turnierplan:

Darstellung eines interaktiven Turnierbaums mit Möglichkeit zur Eingabe von Spielergebnissen.

Grüne Hervorhebung der aktuellen Partie.

Durchgestrichene Darstellung bereits gespielter Partien.

Ergebniseingabe:

Einfache Eingabe der Ergebnisse mit automatischer Fehlerprüfung.

Sicherheitsabfrage beim Überschreiben eines gespeicherten Turniers.

5.3.3. Taskflows

Anmeldung:

Spieler geben ihren Teamnamen ein und gelangen zur Ansicht ihres individuellen Turnierbaums.

Erstellung eines neuen Turniers:

Der Organisator gibt die Teams ein und kann vor Ort Namen ändern.

Eintragung von Ergebnissen:

Ergebnisse werden eingetragen, geprüft, und die nächste Partie wird automatisch hervorgehoben.

Export:

PDF-Export des Turnierplans oder Excel-Export der Ergebnisse.

5.3.4. Bedienbarkeit

Intuitive Navigation und klare Beschriftung der Schaltflächen.

Wenige Buttons, die übersichtlich gestaltet sind.

Entwurfsdokumentation

- Architektur-Notizbuch
- Test Cases

6. Architecture Notebook: Turnierauswertungssoftware

6.1. Links zur Erläuterung

[HTW Seite](#)

[OpenUP Doku](#)

6.2. Zweck

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

6.3. Architekturziele und Philosophie

- Da die Anzahl der Nutzer nur maximal 200 beträgt, wird keine besondere Architektur gebraucht. Es reicht ein einfacher Webserver.
- Die Webseite soll auch offline nutzbar sein und beim späteren synchronisieren eine Möglichkeit bieten, konfligierende Datensätze zu beheben.

6.4. Annahmen und Abhängigkeiten

Annahmen:

- Es wird ein Rechner vom StuRa in der Turnierhalle bereitgestellt, der als Hotspot fungiert, mit dem sich die Spieler verbinden können. Dieser verfügt über eine leistungsstarke Netzwerkkarte, die eine zuverlässige Hotspot-Abdeckung in der gesamten Halle sicherstellt.
- Alle Spieler haben eigene internetfähige Smartphones, auf denen die Software aufgerufen werden kann.
- Unsere eingesetzten Softwarekomponenten laufen weitgehend stabil und werden auch künftig unterstützt.

Abhängigkeiten:

- Wir sind darauf angewiesen, dass der bereitgestellte Rechner des StuRa ohne Verbindungsprobleme läuft, ansonsten funktioniert die Software nicht.

6.5. Architektur-relevante Anforderungen

funktionale Anforderungen:

- Entstehende Daten (Ergebnisse, Turnierhistorie, Teamnamen) etc. müssen gespeichert werden.

- Die Software muss offline funktionieren und Eingaben zwischen verschiedenen Geräten synchronisiert werden.

Benutzbarkeit:

- Die Software muss für den Spieler auf mobilen Endgeräten und für die Organisatoren auf Desktop-Geräten optimiert sein.
- Es muss Mechanismen zum Exportieren von Daten als PDF- und Excel-Datei geben.

Leistung:

- Der Algorithmus zur Turnierplangenerierung muss den vorgegebenen Regeln folgen (Gleichmäßige Aufteilung zwischen Spielen, Pfeifen und Aussetzen, maximal ein Aussetzen nacheinander)

Wartbarkeit:

- Nach Ablauf des Projekts muss die Software von Dritten weiterentwickeln und gewartet werden können.

6.6. Entscheidungen, Nebenbedingungen und Begründungen

Wir haben uns für eine Client-Server-Webanwendung aus folgenden Gründen entschieden:

- Da derselbe Datenbestand von mehreren unabhängigen Benutzern geändert wird, wurde die Datensynchronisation bei einer Peer-To-Peer-Architektur einen zu großen Aufwand erfordern.
- Die Aufwand für die Konfiguration von den Clientgeräten ist gering, wenn zur Nutzung des Softwaresystems auf der Benutzerseite nur ein Webbrowser benötigt wird.

Um bei fehlender Internetverbindung die Systemnutzer kabellos zu vernetzen, muss vor Ort ein WLAN-Netz aufgebaut werden.

Wegen widersprüchlichen Kundenanforderungen an die Clientanzahl, wählen wir zwei Sätze von Architekturkomponenten aus: einmal für 20 Clients und einmal für 200 Clients. Beim Entwurf der Komponentenschnittstellen berücksichtigen wir eine spätere Möglichkeit der Hochskalierung des Systems durch Austausch von Komponenten.

6.6.1. Auswahl der zuverwendenden Komponenten für 20 Clients

1. WLAN-Netzwerk wird mithilfe der in dem Notebook von Organisatoren eingebauten WLAN-Karte gewährleistet und durch das Linux-Betriebssystem verwaltet.
2. Das Software-Produkt wird als eine Java-Anwendung in form von JAR-Datei ausgeliefert.
3. Die Turnierdaten werden zum Archivieren in einfache Dateien geschrieben, beim Programmstart aus den Dateien geladen und intern in den Datenstrukturen von Java (ArrayList, HashMap) verarbeitet.
4. Das Front-End, das die HTML-Code generiert und die Nutzereingaben aus dem Http-Request auswertet programmieren wir selbst.

6.6.2. Auswahl der zuverwendenden Komponenten für 200 Clients

1. Das WLAN-Netzwerk muss aus einem oder mehreren externen WiFi-Access-Points bestehen. Die Einrichtung des Netzwerks gehört nicht zu den Funktionen unseres Softwaresystems.
2. Das Software-Produkt wird mithilfe von [Docker](#) ausgeliefert, damit sichergestellt ist, dass jedes System die Voraussetzungen erfüllt.
3. Zur Speicherung der Daten wollen wir ein Datenbankmanagementsystem nutzen, da die Speicherung in Datenbanken übersichtlicher und robuster ist, als die Speicherung der Daten in Dateien. Desweiteren haben bereits alle Informatik-Studiengänge vorkenntnisse. Beispiel: [PostgreSQL](#)
4. Für die Entwicklung im Front-End könnte ein Framework benutzt werden. Dies würde die Nutzung von API's erleichtern und ein Übersichtliches Routing-System anbieten. Beispiel: [Nuxt](#)
5. Für eine einfachere Gestaltung der Website wird des weiteren [TailwindCSS](#) benutzt.
6. Für das Back-End wird [Java](#) benutzt. Dies vereinfacht die Einarbeitung, da bereits jeder Informatik-Studiengang Java behandelt hat.

6.7. Architekturmechanismen

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

1. Mehrbenutzer-Zugriff.
Gleichzeitige Bedienung der Anwendung durch mehrere Nutzer.
2. Nutzer-Identifizierung.
Eine Folge von Anfragen eines Nutzers, soll demjenigen Nutzer zugeordnet werden können.
3. Daten-Persistenz.
Dienste zum Lesen und Schreiben von gespeicherten Daten.
4. Systemredundanz
Bei Nichterreichbarkeit des Hauptsystems, soll die zweite System die volle Funktionalität übernehmen.

6.8. Wesentliche Abstraktionen

- Das System definiert mehrere wesentliche Abstraktionen, darunter:
 - Benutzer: Personen, die die Anwendung nutzen, insbesondere Turnierleitung und Spieler
 - Turnierplan: Kern des Systems, der die Struktur und Organisation des Turniers abbildet
 - Ergebnisse und Auswertung: Analyse der Ergebnisse und Ermittlung der Gewinner

6.9. Schichten oder Architektur-Framework

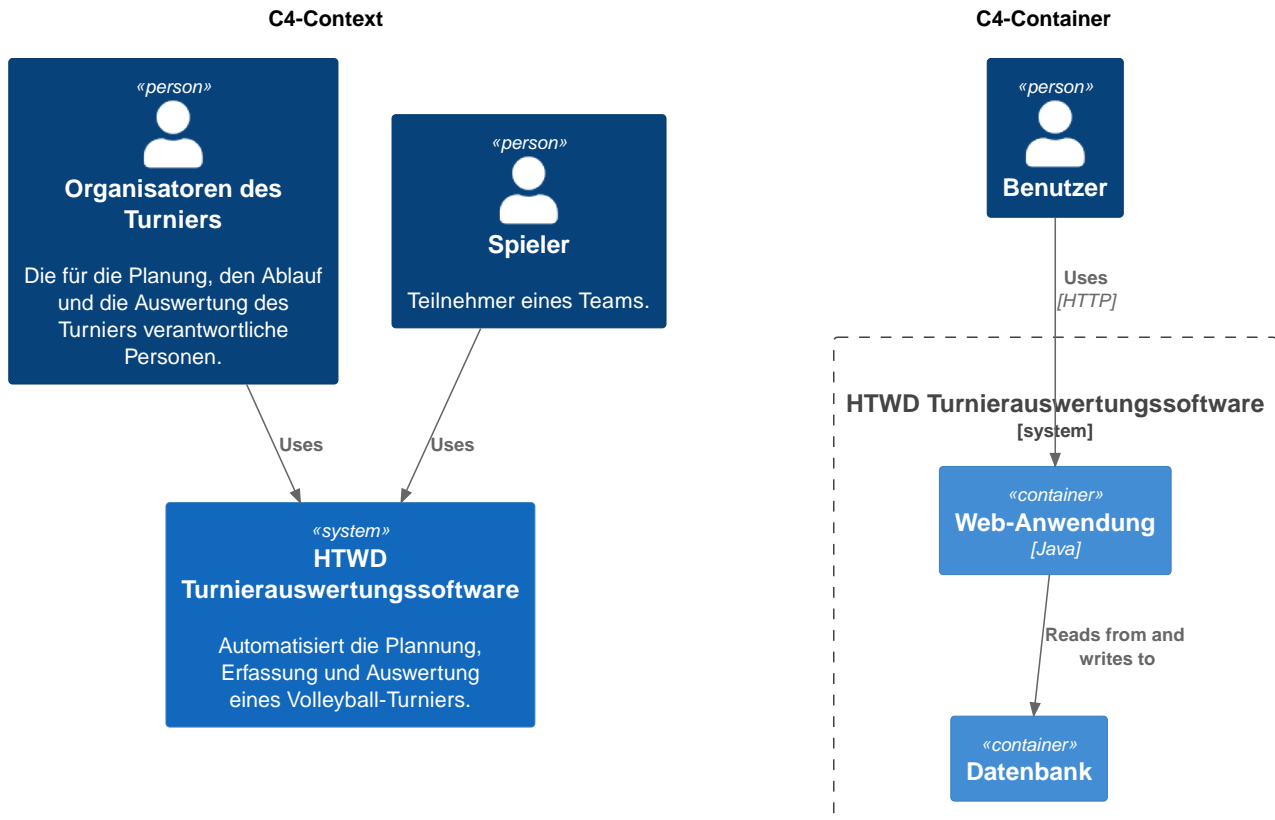
Wir nutzen die Drei-Schichten-Architektur, um eine klare Trennung zwischen den verschiedenen Komponenten der Anwendung zu ermöglichen. Diese Architektur besteht aus folgenden Schichten:

1. **Präsentation** Hier wird die Benutzeroberfläche implementiert. Diese wird dann mittels Front-End-Framework (z.B. Nuxt) erstellt. Diese Schicht ist verantwortlich für die Interaktion mit dem Benutzer und die Darstellung der Daten.
2. **Anwendungslogik** Diese Schicht enthält die Logik der Anwendung. Hier werden die Daten verarbeitet und die Regeln für die Interaktion zwischen der Präsentationsschicht und der Datenbank definiert. Java wird in dieser Schicht verwendet, um die Logik zu implementieren und die Kommunikation mit der Datenbank zu steuern.
3. **Speicherung** Diese Schicht ist für die Interaktion mit der Datenbank verantwortlich. Sie stellt sicher, dass die Daten effizient gespeichert und abgerufen werden. PostgreSQL wird hier als Datenbankmanagementsystem eingesetzt, um die Daten persistent, robust und sicher zu verwalten.

6.10. Architektursichten (Views)

Folgende Sichten werden empfohlen:

6.10.1. Logische Sicht



6.10.2. Physische Sicht (Betriebssicht)

1. Frontend-Server:

- Bereitstellung der Benutzeroberfläche über Nuxt.
- Zugriff über Webbrowser durch Nutzer (Turnierleitung, Spieler).
- Läuft als Docker-Container auf einem Server mit Zugang zum Internet.

2. Backend-Server:

- Ausführung der Anwendungslogik in Java.
- Verwaltung der REST-API, die die Kommunikation zwischen Frontend und Backend steuert.
- Synchronisation von Offline-Daten (falls ein Nutzer offline war und seine Daten später hochlädt).
- Läuft in einem separaten Docker-Container auf demselben oder einem anderen Server wie der Frontend-Server.

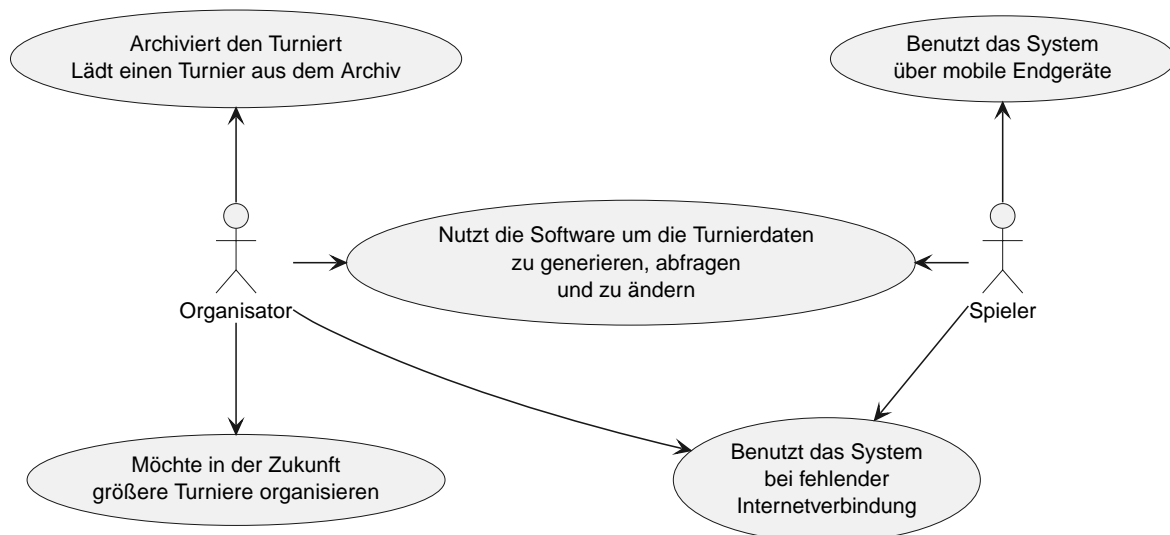
3. Datenbank-Server:

- Speicherung aller persistenten Daten, wie Benutzerprofile, Turnierpläne und Ergebnisse.
- Läuft PostgreSQL als eigenständiger Docker-Container.
- Kann auf einem dedizierten Server gehostet werden, um eine optimale Datenverwaltung zu gewährleisten.

4. Systemkommunikation:

- Alle Komponenten (Frontend, Backend, Datenbank) kommunizieren über ein internes Netzwerk innerhalb des Docker-Ökosystems (z.B. Docker Compose).

6.10.3. Use cases



7. Test Cases: Turnierauswertungssoftware

2025-01-24 21:14:45 +0100 :project-name: Turnierauswertungssoftware :project-system-name: Turnierauswertungssoftware :toc: :toc-title: Inhaltsverzeichnis :toclevels: 2 :sectnums: :icons: font :source-highlighter: rouge :rouge-style: github :xrefstyle: full :figure-caption: Abbildung :table-caption: Tabelle :!example-caption: :experimental: :lang: DE :hyphens: :main-document:

7.1. Allgemeine Informationen

7.2. Test Cases

7.2.1. Test Case 01

7.2.2. Test Case 02