Icon

Description automatically generated**ENTWURF – Projekt „DHBot“**

**Version**: 1.0

**Erstellt am:** 10.11.2021

**Letze Änderung:** 19.11.2021

**Projektgeber:** DHBW Mannheim

DOKUMENTENVERSION

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Nr.** | **Datum** | **Autor** | **Art der Änderung** |
| 0.1 | 10.11.2021 | Daniel Hillmann, Maximilian Brieger, Robin Khatri Chetri, Mathieu Stenzel, Victor Cislari, Tim Hartmann, Sinan Ermis | Erstellung erster Diagramme |
| 0.2 | 17.11.2021 | siehe oben | Erstellung des Dokuments |
| 1.0 | 19.11.2021 | siehe oben | Finalisierung der Version 1.0 |

Inhaltsverzeichnis

[1 Ziele der Architektur 3](#_Toc88515367)

[2 Wesentliche Komponenten 4](#_Toc88515368)

[2.1 Logische Struktur 4](#_Toc88515369)

[2.2 Übersicht des Datenflusses der Applikation 5](#_Toc88515370)

[2.3 Kontrollfluss 5](#_Toc88515371)

[2.4 Physikalische Sicht 7](#_Toc88515372)

[2.5 Anforderungen an die Hardware: 7](#_Toc88515373)

[3 Verwendete Technologien 7](#_Toc88515374)

[3.1 Betriebssystem 7](#_Toc88515375)

[3.2 Datenbanken 8](#_Toc88515376)

[3.3 Programmiersprache 8](#_Toc88515377)

[4 Typische Anwendungsszenarien 9](#_Toc88515378)

[5 Abbildungsverzeichnis 11](#_Toc88515379)

# Ziele der Architektur

Die Architektur ist dafür verantwortlich, die Anforderungen der Applikation festzuhalten. Sie setzt den Rahmen der technischen Möglichkeiten an.  
  
Besonders wichtig ist die hohe Verfügbarkeit der Applikation. Um dies zu gewährleisten, muss die gesamte Anwendung eine hohe Stabilität aufweisen. Daher soll die Applikation selbstheilend sein, sodass falls ein Teil der Anwendung abstürzt, dieser automatisch neu gestartet wird.  
Die Geschwindigkeit wird durch die Wahl taktisch sinnvoller Programmiersprachen wie PHP in Verbindung mit effizienter Kommunikation der einzelnen Komponenten, gewährleistet.

Ebenfalls von großer Bedeutung, ist die Systemunabhängigkeit. Dazu gehört, dass die Applikation virtualisiert ausgeführt wird. Die benötigten Konfigurationsdateien werden mit dem Code mitgeliefert. Es werden keine weiteren Systemkonfigurationen benötigt.

# Wesentliche Komponenten

## Logische Struktur

In Abbildung 1 kann die Struktur des Programmes mit den Schnittstellen zwischen den verschiedenen Programmteilen aufgezeigt.



Abbildung 1: Übersicht über die Komponenten und deren Schnittstellen

## Übersicht des Datenflusses der Applikation



Abbildung 2: Übersicht des Datenflusses der Applikation

In Abbildung 2 werden Komponenten der Software verschiedenen Schichten zugeordnet und der Datenfluss zwischen diesen Komponenten wird aufgezeigt.

## Kontrollfluss

In den folgenden Abbildungen wird aufgezeigt wie bei Nachrichtenempfang der Ablauf innerhalb des Programms aussieht.



Abbildung 3: Darstellung des parallelen Ablaufs von Aktionen bei dem Empfang von 2 Nachrichten

****

Abbildung 4: Methodenaufrufe bei dem Empfang einer Nachricht

## Physikalische Sicht

Die Applikation wird vollständig serverseitig ausgeführt. Die Hardware kann dabei - oberhalb der Minimalforderung - frei gewählt werden und durch die Benutzung von Docker muss keine manuelle Installation von DBMS oder PHP erfolgen.



Processing-Unit

Abbildung 5: Überblick über das Deployment

## Anforderungen an die Hardware:

In der physischen Schicht muss ein Host-System zur Verfügung stehen, das in der Lage ist eine Docker-Instanz zu betreiben. Der Betrieb von PHP und PostgreSQL sehen wir mindestens 2 Kerne und 2 GB RAM vor. Für eine hohe Verfügbarkeit des Bots muss die Internetverbindung im Betrieb hochverfügbar (>98%) sein.

# Verwendete Technologien

## Betriebssystem

Als Betriebsgrundlage wurde Docker gewählt. Docker ist eine freie Software zur Container-Virtualisierung. Durch Docker lassen sich Anwendungen leicht mit allen nötigen Paketen zu einem Container verpacken.

Weil sich der Container leicht transportieren und einbinden lässt, ermöglicht das ein einfaches Deployment und eine einfache Entwicklung der Anwendung. Weiter lassen sich die Ressourcen unkompliziert erweitern, indem dem Container mehr Ressourcen zugewiesen oder weitere Container gestartet werden. Aufgrund dessen, dass die Docker Virtualisierung für jedes Betriebssystem verfügbar ist, entsteht eine sehr hohe Kompatibilität.

## Datenbanken

Als Datenbank setzen wir auf die PostgreSQL Datenbank. PostgreSQL ist ein freies, Open-Source objektrelationales Datenbankmanagementsystem. In Abbildung 6 findet sich das Datenbankmodell mit allen benötigten Tabellen und Beziehungen.



Abbildung 6: Übersicht über die Tabellen in der Datenbank

## Programmiersprache

Für das Projekt soll die Sprache PHP benutzt werden.

PHP wurde gewählt, da dieses Projekt durch die Anbindung an den Vorlesungsplan und das Dualis-System einen hohen Anteil an Web-Parsing enthält, welcher mit der vorhandenen Expertise am einfachsten in PHP umsetzbar scheint.

Außerdem unterstützt PHP gängiges objektorientiertes Programmieren, welches es ermöglicht die vielen unterschiedlichen Teilaufgaben zu modularisieren.

# Typische Anwendungsszenarien



Abbildung 7: Darstellung des allgemeinen Use Cases

Im Folgenden werden vier Beispiele der Bot-Kommunikation aufgeführt. Mögliche Stichwörter, die die Software erkennen soll, sind in Fett-Schrift geschrieben:

**Beispiel 1:** Wie sieht der **Stundenplan heute** aus?

Datenquelle: “vorlesungsplan.dhbw-mannheim.de”

Die Vorlesungen des Tages werden zu einer Telegram-Nachricht umgeformt.

Beispiel für eine Antwort:

Folgende Vorlesungen hast du heute noch:

09:30 – 14:00 - Software Engineering

16:00 – 17:30 – Assembler

**Beispiel 2:** Welche **Klausuren stehen** noch an?

Datenquelle: “vorlesungsplan.dhbw-mannheim.de”

Bezogen auf das derzeitige Datum werden die verbliebenen anstehenden Klausuren für das derzeitige Semester angegeben.

Beispiel für eine Antwort:

Folgende Klausuren musst du dieses Semester noch schreiben:

2021.12.21 - 09:30 – 11:00 – Theoretische Informatik Formale Sprachen

2021.12.22 - 09:30 – 11:00 – Datenbanken

**Beispiel 3:** Welche **Noten** habe ich in den **Klausuren** aus dem **2. Semester** geschrieben?

Datenquelle: “dualis.dhbw.de”, Noten der Module im angegebenen Semester

Beispiel für eine Antwort:

Folgende Noten hast du im 2. Semester erreicht:

* Mathematik I: 1,0
* Theoretische Informatik II: 1,0
* Programmieren: 1,0
* Technische Informatik I: 1,0
* Anwendungsprojekt Informatik: 1,0

**Beispiel 4:** Wie viele **Vorlesungen** habe ich bis zur **Klausur** im Modul **Betriebssysteme**?

Datenquelle: “vorlesungsplan.dhbw-mannheim.de”

Beispiel für eine Antwort:

Du hast noch 3 Vorlesungen mit insgesamt 9 Stunden bis zur Klausur in Betriebssysteme.

# Abbildungsverzeichnis

[Abbildung 1: Übersicht über die Komponenten und deren Schnittstellen 4](#_Toc88230129)

[Abbildung 2: Übersicht des Datenflusses der Applikation 5](#_Toc88230130)

[Abbildung 3: Darstellung des parallelen Ablaufs von Aktionen bei dem Empfang von 2 Nachrichten 6](#_Toc88230131)

[Abbildung 4: Methodenaufrufe bei dem Empfang einer Nachricht 6](#_Toc88230132)

[Abbildung 5: Überblick über das Deployment 7](#_Toc88230133)

[Abbildung 6: Übersicht über die Tabellen in der Datenbank 8](#_Toc88230134)

[Abbildung 7: Darstellung des allgemeinen Use Cases 9](#_Toc88230135)