**IT-Konzept**

**Projektname: Weichware – PDF2Hörbuch**

**Version:** 0

**Autor:** Friese, Weichware

Dokumenthistorie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aufgabe | Termin | Autor | Status | Version |
| IT-Konzept erstellen | 18.10.2022 | Friese | Erledigt | 0 |
| Bedienoberfläche erstellen | Offen |  |  |  |
| Serverstruktur erstellen | Offen |  |  |  |
| PHP-Programm  Upload | Offen |  |  |  |
| PHP-Programm Textausgabe-API | Offen |  |  |  |
| PHP-Programm  Userausgabe | Offen |  |  |  |
| PHP-Programm  Nutzeraccount | Offen |  |  |  |
| Abnahme für  Projektstart | Offen |  |  |  |
| Übersetzung in weitere Sprachen | Offen |  |  |  |
|  |  |  |  |  |

Inhaltsverzeichnis

[**1.** **Gesamtüberblick** 2](#_Toc10980)

[**2.** **Basis** 2](#_Toc10981)

[**2.1.** **Überblick** 2](#_Toc10982)

[**2.2.** **Architektur** 2](#_Toc10983)

[**2.2.1.** **Anwendungsdiagramm** 2](#_Toc10984)

[**2.3.** **Voraussetzungen** 3](#_Toc10985)

[**2.3.1.** **Software** 3](#_Toc10986)

[**2.3.2.** **<Hardware>** 3](#_Toc10987)

[**2.3.3.** **<Daten>** 3](#_Toc10988)

[**2.4.** **Zielsetzungen** 3](#_Toc10989)

[**2.4.1.** **<Software>** 3](#_Toc10990)

[**2.4.2.** **<Performanz>** 4](#_Toc10991)

[**2.4.3.** **<Verfügbarkeit>** 4](#_Toc10992)

[**2.5.** **Ansprechpartner** 4](#_Toc10993)

[**2.5.1.** **<fachlich>** 4](#_Toc10994)

[**2.5.2.** **<technisch>** 4](#_Toc10995)

[**3.** **Daten** 4](#_Toc10996)

[**3.1.** **Überblick** 4](#_Toc10997)

[**3.2.** **Architektur** 4](#_Toc10998)

[**3.2.1.** **Datenflussdiagramm** 5](#_Toc10999)

[**3.3.** **Datenentitäten** 5](#_Toc11000)

[**3.4.** **<Datenintegrität>** 5](#_Toc11001)

[**3.5.** **<Datenmodelle>** 5](#_Toc11002)

[**4.** **Steuerung** 6](#_Toc11003)

[**5.** **Verarbeitung** 6](#_Toc11004)

[**5.1.** **Überblick** 6](#_Toc11005)

[**5.2.** **Architektur** 6](#_Toc11006)

[**5.2.1.** **< Zugriffsdiagramm>** 6](#_Toc11007)

[**5.2.2.** **Klassenmodell** 6](#_Toc11008)

[**5.3.** **Verarbeitungsentitäten** 7](#_Toc11009)

[**5.4.** **<Datenplausibilität>** 7](#_Toc11010)

[**6.** **Protokollierung** 7](#_Toc11011)

[**7.** **Qualitätssicherung** 7](#_Toc11012)

[**7.1.** **<Programmierrichtlinien>** 7](#_Toc11013)

[**7.2.** **Test** 7](#_Toc11014)

# Gesamtüberblick

Ziel des Projektes ist es für Studenten eine Anwendungssoftware zu schaffen, welche es ermöglicht Skripte, Artikel und weitere Textdokumente als Audioformat auf mobilen Geräte zur Verfügung zustellen

# Basis

Dieses Projekt wird im Rahmen einer Seminaraufgabe im Wintersemester 2022/2023 erstellt und bearbeitet.

## Überblick

**Allgemeine Informationen:**

Eingabe von Text

Umwandlung mit TTS

Ausgabe von Audio

**Anforderungen:**

**Allgemeine Bedingungen:**

PHP

HTML

Javascript

## Architektur

Besondere Bemerkungen zur Architektur:

Schnittstellen als Webservice

API TTS

### Anwendungsdiagramm

Grafische Gesamtdarstellung aller beteiligten Systeme

PHP-Server , API , Browser

welche Schnittstellen und Daten

Stichpunkte für den Diagramminhalt:

Daten

Anwendungen

<Schnittstellen>

<Fremdsysteme>

## Voraussetzungen

Programmierkenntnisse in PHP,

Texttospeech-Umwandlung

### Software

Welche Software muss auf welchen Systemen vorhanden sein.

Benutzer: Browser

System: PHP-Server

Extern: TTS-Api

### <Hardware>

PHP-Server

### <Daten>

Text

Evtl. Benutzerkonto

## Zielsetzungen

Beschreibung allgemeiner Ziele wie z.B. dass der Schwerpunkt auf Wartbarkeit oder auf Sicherheit zu setzen ist.

### <Software>

Das Webinterface soll auf den aktuellen Version des FireFox Klar und MS Edge und Chrome Mobil lauffähig sein.

Die Datenschnittstellen sollen wo immer möglichst CSV verwenden.

Die Anwendungen müssen auf Windows10 und Android laufen.

### <Performanz>

Die Webanwendung darf eine maximale Zugriffszeit von 30 Sekunden bei 10 parallelen Nutzern nicht überschreiten.

### <Verfügbarkeit>

Die Anwendung wird mehrmals im Semester benötigt.

## Ansprechpartner

### <fachlich>

<Konzeption>

<Organisation>

### <technisch>

<Administration>

<Entwicklung>

# Daten

Eine allgemeine Beschreibung wie:

Mit Hilfe der Anwendung werden täglich Kundendaten aufgenommen, in Berichtsform zur Verfügung gestellt sowie per Nachtlauf strukturiert an andere Zielsysteme weitergegeben.

## Überblick

Allgemeine Informationen oder Anforderungen.

Hier können Informationen z.B. zur Berücksichtigung der schlechten Datenqualität oder zu besonderen Datenquellen beschrieben werden.

Weitere Beispiele:

Alle Zeichenketten sollen stets im Unicode verwendet werden.

Die Genauigkeit der Nachkommastellen muss mindestens 8 betragen.

Länderspezifische Datums- oder Zeitwerte sollen vor jeglicher Verarbeitung jeweils in ein internes Datumszeitformat gewandelt werden.

## Architektur

Hier könnte eine zusätzliche Beschreibung der gewünschten Architekturansätze stehen.

### Datenflussdiagramm

In einem solchen Diagramm sollten die folgenden Themen Berücksichtigung finden:

<Konfiguration>

<Datenbanken>

<Textdateien>

<Datenströme>

<Anwenderinteraktion>

<Anwendungen>

## Datenentitäten

Hier können die Anforderungen an die einzelnen Datenstrukturen und -pakete, welche im Diagramm unter 3.2.1 bereits aufgezeigt wurden, detailliert beschrieben werden.

Beispiel:

Die CSV-Tabelle XY muss jeden Tag um 18 Uhr in Ordner XYZ abgelegt werden.

Was

Datentypen

Datenformate

<Datenbereiche>

Woher

<Siehe 3.2.1.>

Wohin

<Siehe 3.2.1.>

Wann

<Aktualisierungen>

<Intervalle>

## <Datenintegrität>

Beispiel:

Es dürfen nur Marktdaten verarbeitet werden, zu denen auch eine Währung hinterlegt ist.

<Annahmen>

<Prüfungen>

## <Datenmodelle>

Hier sollten z.B. die Datenmodellierungen zu verwendeten Datenbanken per Skript, Diagramm und oder Referenz auf externe Dateien hinterlegt werden.

Aber auch ein paar Exceltabellen oder z.B. XML-Dateien sind es gelegentlich wert hier ordentlich modelliert zu werden.

# Steuerung

Hier soll beschrieben werden, wie, von wem und wann genau eine Anwendung aufgerufen werden soll. Eine einfache Rechteverwaltung kann hier mit beschrieben werden. Für eine komplexere Rechteverwaltung ist eher ein eigenes Kapitel empfehlenswert.

Was

<Anwender>

<Berechtigungen>

<Anwendungen>

<Berechtigungen>

<Services>

<Berechtigungen>

Wann

<Termine>

<Intervalle>

Wie

<Oberfläche>

<Parameter>

# Verarbeitung

Beschreibung allgemeiner Anforderungen an die Verarbeitung wie z.B. dass der Schwerpunkt auf Wartbarkeit oder auf Sicherheit zu setzen ist.

## Überblick

Allgemeine Informationen oder Anforderungen.

Beispiele:

Alle Anwendungsschnittstellen dürfen ausschließlich Datentypen verwenden, welche auch direkt mit J2EE ansprechbar bzw. verwendbar sind.

## Architektur

Beschreibung von Besonderheiten der Anwendungsarchitektur.

### < Zugriffsdiagramm>

<Anwendungsschichten>

Hier ein Beispiel unter http://de.wikipedia.org/wiki/Model\_View\_Controller

### Klassenmodell

Ähnlich eines Datenmodells werden hier die Klassen bzw. Module einer Anwendung beschrieben mit Hilfe von

<Klassen- bzw. Moduldiagramm>

oder auch noch etwas detaillierter in einem

<Funktions- bzw. Prozedurdiagramm>

## Verarbeitungsentitäten

Hier können einzelne Strukturen aus den Diagrammen unter 5.2.2 gesondert detaillierter beschrieben werden.

Was

<Module>

<Klassen>

<Funktionen>

<Prozeduren>

Wo

<Dateien>

<Netzwerk>

Wann

Startverhalten

## <Datenplausibilität>

Beispiel:

Es dürfen nur Marktdaten verarbeitet werden, deren Abweichung zum Vortag nicht mehr als 10% beträgt.

<Annahmen>

<Prüfungen>

# Protokollierung

Die Protokollierung der technischen und fachlichen Protokolle wird von Jan Friese übernommen. Die Protokolle des Projektmanagement werden von Parik Grass übernommen.

# Qualitätssicherung

Allgemeine Informationen oder Anforderungen.

## <Programmierrichtlinien>

## Test

HTML-Test

Benutzertest

100 Testfälle mit verschiedenen Texten und Skripten