

Pflichtenheft

Projektbezeichnung	AC DC
Projektleiter	Pascal Tomek, Konrad Mädler
Erstellt am	10.10.2023
Letzte Änderung am	11.12.2023
Status	In bearbeitung
Aktuelle Version	1.2

Änderungsverlauf

Nr.	Datum	Version	Geänderte Kapitel	Art der Änderung	Autor	Status
1	12.10.2023	1.0	Alle	Erstellung	Konrad Mädler	Abgeschlossen
2	17.11.2023	1.1	Zeitplan, Technische Anforderungen, Funktionale Anforderungen	Bearbeitung	Konrad Mädler, Pascal Tomek	Abgeschlossen
3	11.12.2023	1.2	Teams. Funktionale & Nicht-Funktionale Anforderungen, Zeitplan, Inhaltsverzeichnis , Formatierung	Bearbeitung	Pascal Tomek	Abgeschlossen

1 Einleitung	2
2 Allgemeines	2
2.1 Ausgangssituation	2
3 Konzept	3
3.1 Ziele des Anbieters	3
3.2 Ziele und Nutzen des Anwenders	3
3.3 Zielgruppe	4
4 Anforderungen	4
4.1 Funktionale Anforderungen	4
4.1.0 Startseite	4
4.1.1 Benutzer-Dashboard	4
4.1.2 Registrierung und Login für Nutzer	4
4.1.3 Benutzerprofil	4
4.1.4 Listenansicht	5
4.1.5 Lernfunktion	5
4.1.6 Suchfunktion	5
4.1.7 Begriffs Ansicht	5
4.1.8 Datenbank	5
4.1.9 Ordnerfunktion	5
4.2 Technische Anforderungen	6
4.2.0 Server	6
4.2.1 Datenbank	6
4.2.2 Kompatibilitäts Anforderungen	6
4.2.3 Sicherheit	6
4.2.4 Performance	7
4.2.5 Tech-Stack	7
4.3 Nicht-funktionale Anforderungen	7
4.3.1 Allgemeine Anforderungen	7
4.3.1.1 Benutzer Dashboard	7
4.3.1.2 Registrierung und Login für Nutzer	7
4.3.1.3 Benutzerprofil	8
4.3.1.4 Listenansicht	8
4.3.1.5 Lernfunktion	8
4.3.1.6 Suchfunktion	8
4.3.1.7 Begriffs Ansicht	8
6 Rahmenbedingungen	8
6.0.1 Gesamte Bearbeitungszeit	8
6.0.2 Arbeitsweise	8
6.1 Zeitplan	9
6.2 Problemanalyse	10
7 Liefer- und Abnahmebedingungen	11
8 Skizzen des Frontends	12

1 Einleitung

Das vorliegende Pflichtenheft enthält die an das zu entwickelnde Produkt gestellten funktionalen sowie nicht-funktionalen Anforderungen. Es dient als Basis für die Ausschreibung und Vertragsgestaltung und bildet somit die Vorgabe für die Angebotserstellung. Kommt es zwischen Auftragnehmer und Auftraggeber zu einem Vertragsabschluss, ist das bestehende Pflichtenheft rechtlich bindend. Alle zuvor zwischen Auftraggeber und Auftragnehmer getroffenen Absprachen verlieren in der Regel durch das Pflichtenheft ihre Gültigkeit – sofern hier nichts Gegenteiliges vermerkt ist. Mit den Anforderungen werden die Rahmenbedingungen für die Entwicklung festgelegt, die vom Auftragnehmer im Pflichtenheft detailliert ausgestaltet werden.

Die zu entwickelnde Web-Anwendung (i.F. "ACDC 1.0") wird für das Lernen und das Nachschlagen von Begriffen Verwendung finden, außerdem wird sie über jeden Browser zugänglich sein. Es ist essenziell, dass die Anwendung eine rundum abgestimmte und sichere Plattform wird, um eine Bereicherung des Unternehmens werden zu können.

2 Allgemeines

2.1 Ausgangssituation

Momentan werden Begriffe der Informatik und Software-Entwicklung in einer unübersichtlichen Tabelle im Confluence hinterlegt, welche Azubis für relevant halten. Die Pflege und Erweiterung der Tabelle wurde über die Jahre seitens der Azubis vernachlässigt. Die Abfrage und Wiederholung solcher berufsrelevanten Begriffe und Konzepte erfolgt meist in Wissenstests und nimmt somit Zeit in Anspruch, die zur Vermittlung neuer Themen genutzt werden könnte.

2.2 Teams

Rolle(n)	Name	Telefon	E-Mail	Team
Projektbetreuer	Falk Döring		falk.doering@check24.de	Core-PHP/ Ausbilder
Entwickler	Pascal Tomek		pascal.tomek@check24.de	Azubi Entwicklerteam
Entwickler	Konrad Mädler		konrad.maedler@check24.de	Azubi Entwicklerteam

3 Konzept

3.1 Ziele des Anbieters

Das Hauptziel des Anbieters ist es, eine benutzerfreundliche und effiziente Lern- und Nachschlageplattform namens "ACDC 1.0" zu entwickeln, die den Anwendern dabei hilft, berufsrelevante Begriffe und Konzepte leichter zu erlernen und nachzuschlagen. Dabei sollen folgende Ziele verfolgt werden:

- Die Schaffung einer leicht zugänglichen Browser basierten Anwendung mit einer intuitiven Benutzeroberfläche.
- Die Förderung des aktiven Lernens durch die Bereitstellung von Lernfunktionen.
- Die Verbesserung der Zusammenarbeit zwischen den Nutzern, indem sie Begriffe vorschlagen und gemeinsam an der Pflege der Inhalte arbeiten können.

3.2 Ziele und Nutzen des Anwenders

Die Ziele und der Nutzen für die Anwender von "ACDC 1.0" sind eng mit den funktionalen Anforderungen verknüpft, die im nächsten Abschnitt beschrieben werden. Zu den Hauptzielen der Anwender gehören:

Die Möglichkeit, leicht auf relevante Begriffe und Definitionen zugreifen zu können, um ihr Verständnis zu vertiefen.

Die Option, eigene Begriffe und Definitionen hinzuzufügen und von anderen Nutzern überprüfen zu lassen.

Chance, Lernmaterial individuell anzupassen und effizienter zu lernen.

3.3 Zielgruppe

Die Zielgruppe für "ACDC 1.0" sind Azubis in verschiedenen Lehrjahren, die nach einer einfachen und effektiven Möglichkeit suchen, berufsrelevante Begriffe und Konzepte zu erlernen. Die Zielgruppe zeichnet sich durch folgende Merkmale aus:

- Ein Interesse an der Software-Entwicklung und Informationstechnologie.
- Die Bereitschaft zur Zusammenarbeit und zum Teilen von Wissen.
- Ein breites Altersspektrum, da sowohl junge als auch erfahrene Azubis von der Plattform profitieren sollen.

4 Anforderungen

4.1 Funktionale Anforderungen

4.1.0 Startseite

Eine Homepage mit einem Login-Formular für E-Mail und Passwort, das durch einen Anmeldeknopf bestätigt werden muss.

Siehe Abb.: 1

4.1.1 Benutzer-Dashboard

Ein Benutzer-Dashboard, das standardmäßig das Benutzerprofil darstellt und ein Menü mit den folgenden Elementen enthält:

- Suchfunktion
- Lernfunktion
- Listenansicht (A-Z)
- Vorschlägen von Begriffen – Benutzerprofil

Siehe Abb.: 2

4.1.2 Registrierung und Login für Nutzer

Um das Erstellen von Nutzern zu automatisieren, wird ein Registrierungsformular implementiert. Dies sollte den aktuellen Sicherheitsstandards entsprechen und den Anforderungen des Unternehmens gerecht werden, ebenso der Login. Beides erfolgt über ein separates Formular und sollte branchenübliche Funktionen bieten (Passwort vergessen? etc.).

Siehe Abb.: 1

4.1.3 Benutzerprofil

Ein Benutzerprofil, welches folgende Elemente/Inhalte darstellt:

- E-Mail
- Name
- Statistiken(optional)
- Ordnerliste mit Knopf zum Hinzufügen eines Ordners

Siehe Abb.: 4

4.1.4 Listenansicht

Eine Listenansicht, die alle Begriffe und Definitionen der in der Datenbank liegenden Einträge alphabetisch sortiert darstellt. Am Kopf der Darstellung ist die Suchfunktion implementiert.

Siehe Abb.: 5

4.1.5 Lernfunktion

Eine Lernfunktion, die eine Auswahl verschiedener Modi zum Lernen der Karteikarten (Begriffe & Definitionen) darstellt. Es sollten mindestens 2 Modi verfügbar sein: Begriff zu Definition und Definition zu Begriff. Dabei kann ein bestimmter Ordner zum Lernen gewählt werden. Jede Darstellung einer Karteikarte beinhaltet ein Text-Eingabefeld zur Selbstüberprüfung.

Siehe Abb.: 6

4.1.6 Suchfunktion

Eine Suchfunktion, welche nach Texteingabe alle passenden Einträge(Begriffe & Definitionen) der Datenbank in einer Listenansicht ausgibt.

4.1.7 Begriffs Ansicht

Eine Begriffs Ansicht, welche einen einzelnen Eintrag aus der Datenbank darstellt. Ein Knopf zur Markierung für die Lernfunktion, welche ein Drop-down-**Menü** für die Ordner öffnet, in der der Eintrag hinterlegt werden soll.

Siehe Abb.: 7

4.1.8 Datenbank

Eine Datenbank, welche Nutzer, deren Ordner und Markierungen und Begriffe und Definitionen speichert.

4.1.9 Ordnerfunktion

Eine Ordnerfunktion, welche das Erstellen, Umbenennen und Löschen von Ordnern ermöglicht.

4.2 Technische Anforderungen

4.2.0 Server

Ein Server ist notwendig, um die Daten zu verarbeiten, zu speichern, die Kommunikation zu ermöglichen, Sicherheit zu gewährleisten und den Zugriff auf Anwendungen und Ressourcen zu steuern, was für die Arten von Anwendungen unerlässlich ist.

Es sollte vor Integration der Anwendung auf bestehende Server geprüft werden, ob die Anforderung und die Kapazität ausreichend sind.

Empfohlene Anforderungen:

Betriebssystem: Linux Debian

CPU: Vierkern-Prozessor (Cache Kapazität abhängig von der Anzahl von Nutzern) RAM: 8 GB

Speicher: 500 GB HDD

4.2.1 Datenbank

Wir werden mit Laminas als Framework (externes Toolkit für eine bestehende Programmiersprache) für PHP arbeiten, in diesem ist eine kostenlose Datenbank-Komponente enthalten und wir werden diese für den Auftrag nutzen um in einer MySQL- Datenbank Die gespeicherten Definitionen, Begriffe, Nutzer IDs, Markierungen, Passwörter und Benutzernamen/ E-Mails zu speichern.

4.2.2 Kompatibilitäts Anforderungen

Es ist wichtig, dass die Web-Anwendung in allen Browsern Funktionale- und Nicht-Funktionale Anforderungen bereitstellt, deshalb binden wir Tests für das automatische Prüfen des Frontend (Der sichtbare Teil der Webanwendung) ein, dafür brauchen wir Tools wie zum Beispiel Rainforest QA.

Natürlich werden manuelle Tests in verschiedenen Browsern zur Entwicklung der Anwendung beitragen.

4.2.3 Sicherheit

Validieren der User Eingaben

Es ist essentiell, dass die Anwendung sicher vor spezifisch auf die Anwendung abgestimmte Angriffe ist. Ein mögliches Angriffsszenario wäre:

SQL-Injektionen(XSS): Das Angreifen der Datenbank-Komponente der Anwendung über Code-Injektionen. Dies kann bei allen Eingabeaufforderungen/Möglichkeiten geschehen, daher müssen die Eingaben streng validiert werden, um den Betrieb und die personenbezogenen Daten in Sachen Integrität und Diebstahl zu schützen.

4.2.4 Performance

Die Performance der Anwendung ist wichtig, um ein angenehmes Nutzen der Anwendung möglich zu machen. Wir setzen daher auf Tests, die darauf abgestimmt sind, die Ladezeiten zu überprüfen. Unser Ziel ist es, die Anwendung mit Ladezeiten von unter 1,5 Sekunden zu erreichen. Dies könnten wir durch das Vorladen der User bezogenen Daten erreichen.

Zudem führen wir Lasttests durch, um unsere Anwendung denen anzupassen und falls nötig eine Orchestrierung des Servers zu implementieren.

4.2.5 Tech-Stack

Das Projekt soll einen bildenden Zweck für das Entwicklerteam beinhalten. Um dies zu gewährleisten, beschränkt sich die zu Nutzenden Technologien auf die hier genannten. Sollten Technologien benötigt werden, die außerhalb des definierten Tech-Stacks liegen, so ist dies mit der Projektbetreuung abzusprechen und nachträglich zu ergänzen.

- Programmiersprache: PHP
- Framework: Laminas
- SQL-Dialekt: MySQL
- Webserver: Apache
- NoSQL-Management-Software: Redis
- Containerisierung: Docker
- Deployment: Jenkins

4.3 Nicht-funktionale Anforderungen

4.3.1 Allgemeine Anforderungen

4.3.1.1 Benutzer Dashboard

Menü befindet sich im Kopf der Webseite:

Ansicht mit Links von Links nach Rechts:

Logo1 , Listenansicht, Lernfunktion, Space bis ganz Rechts, Suchfunktion mit Lupen-Emoji, Benutzerprofil

4.3.1.2 Registrierung und Login für Nutzer

Eingabemaske zentriert, Textfelder mit E-Mail & Passwort, Absenden-Knopf unterhalb der Maske

4.3.1.3 Benutzerprofil

Übersicht auf Webseite zentriert, Vorname & Nachname, E-Mail, Statistiken(optional), Ordnerliste mit Knopf zum Hinzufügen eines Ordners

4.3.1.4 Listenansicht

Darstellung aller Begriffe und Definitionen der in der Datenbank liegenden Daten alphabetisch sortiert als Tabelle, beinhaltet Suchfunktion über der Tabelle, Definition wird auf Zeichenanzahl dynamisch begrenzt(evtl. nach Auflösung), Begriffe sind als href-Links dargestellt um auf Begriffs Ansicht zuzugreifen

4.3.1.5 Lernfunktion

Startseite verfügt über Modi-Knöpfe mit Ref-Links zu jeweiligen Modi, Darstellung der Begriffe/Definition als Karteikarte, Eingabe zur Selbstüberprüfung unterhalb der Karteikarte mit Bestätigen Knopf hinter Eingabe-Textfeld

4.3.1.6 Suchfunktion

Textfeld mit Lupen-Emoji, nach Bestätigung führt dies zur Listenansicht mit allen passenden Begriffen, konform zur Listenansicht.

4.3.1.7 Begriffs Ansicht

Überschrift = Begriff, Paragraph = Definition, neben Überschrift Button zur Markierung(Drop-down Menü mit Ordnern) Mit Stern-Emoji, der gefüllt wird, sobald es für die Lernfunktion markiert wurde.

6 Rahmenbedingungen

6.0.1 Gesamte Bearbeitungszeit

Die Zeit der Entwicklung der Anwendung wird 13 Wochen beanspruchen. Das Azubi-Entwicklungsteam wird täglich 6 Stunden daran arbeiten. Es wird mit einer drei Wochen Pufferzeit gerechnet, die durch fehlendes Personal oder unerwartete Ereignisse in Anspruch genommen werden kann.

6.0.2 Arbeitsweise

Das Team wird nach dem Kanban-Ansatz arbeiten, dieser Ansatz zeichnet sich durch fest definierte Phasen aus, durch die man als Team läuft. Dies bietet sich bei einem fest definierten Produkt an.
test driven, Domain Driven

6.1 Zeitplan

Phase 1: Planung

- Woche 1:
 - Projektstart und Kick-off-Meeting
 - Ticketierung
 - Systementwurf
 - Eventstorming
 - Domäne ausarbeiten
 - Planung der Datenbank
 - Planung der Ordner- & Modulstruktur

Phase 2: Umsetzung Prototyp

- Woche 2:
 - Erstellen der Entwicklungsumgebung
 - Container mit PHP, Laminas Framework, XDebug, MySQL, Apache, Redis
 - Erstellen der Ordner- & Modulstruktur
 - Umsetzung des Prototypen
 - Karteikarten anzeigen
 - Listenansicht aller Karteikarten
 - Lern-ansicht

Phase 3: Umsetzung MVP

- Woche 3-4:
 - Umsetzung des MVP
 - Import von Karteikarten als CSV
 - Login & Registrierung

Phase 4: Feedback einholen und umsetzen

- Woche 5:
 - Vorstellung bei Projektverantwortlichen
 - Präsentation der Funktionalitäten
 - Feedback zur Optimierung einholen
 - Feedback umsetzen

Phase 5: Weitere Features

- Woche 6-9:
 - Weitere Features umsetzen:
 - Benutzerprofil
 - Benutzer-Dashboard
 - Lernmodi
 - Kategorien
 - benutzerspezifische Ordner/Listen
 - Suchfunktion
 - Hinzufügen & Bearbeiten von Karteikarten

Phase 6: Feedback einholen und umsetzen

- Woche 10-11:
 - Vorstellung bei Projektverantwortlichen
 - Präsentation der Funktionalitäten
 - Feedback zur Optimierung einholen
 - Feedback umsetzen

Phase 6: Abnahme und Retrospektive des Projekts

- Woche 10-11:
 - Vorstellung bei Projektverantwortlichen
 - Präsentation der Funktionalitäten
 - Feedback zum gesamten Projekt einholen
 - Retrospektive planen und abhalten
 - Freigabe zum Release

Phase 7: Veröffentlichung auf Live Umgebung

- Woche 12:
 - Installation von Docker auf Live-Server
 - Automatisierung der kontinuierlichen Integration und Veröffentlichung
 - Veröffentlichen von ACDC 1.0 auf die Live Umgebung

Phase 7: Wartung der Software und Pflege der Datenbank

- ab Woche 13:
 - Einrichtung eines Server Monitorings
 - Regelmäßige Kontrolle des Server Monitoring
 - Beginn der Wartungsphase, Fehlerkorrekturen, Aktualisierungen und laufender Support

Achtung: Die angegebenen Zeiten sind nur grobe Richtwerte und können abweichen.

6.2 Problemanalyse

Fehlende Kompetenzen

Durch fehlende Kompetenzen kann es zu zeitlichen Problemen kommen, da viele Inhalte, die relevant für die Entwicklung der Anwendung sind, noch erlernt werden müssen.

Es kann zu Fehlern in der Architektur der Codebase kommen, die uns erst sehr spät im Entwicklungsprozess auffallen könnten.

7 Liefer- und Abnahmebedingungen

1. Die Web-Anwendung ist von allen gängigen Browsern erreichbar und die Funktionalität ist nicht beeinträchtigt

2. Ein Nutzer kann sich mit E-Mail und Passwort einloggen und wird zum Benutzer-Dashboard weitergeleitet

3. Die Listenansicht zeigt alle in der Datenbank liegenden Einträge an und wird mit einer maximalen Anzeige ausgestattet welche durch Paginierung weiterer Einträge erreichbar macht

4. Die Suchfunktion zeigt sowohl alle Begriffe als auch alle Definitionen welche den Suchbegriff enthalten in einer Listenansicht an

5. Das Hinzufügen eines Begriffes löst einen Pull-Request bei einem Hauptentwickler und einem Ausbilder aus. Nach erfolgreichem Bestätigen wird dieser in die Datenbank übernommen

6. Die Lernfunktion unterstützt beide Lern-Modi und gibt in zufälliger Reihenfolge Begriffe aus dem ausgewählten Ordner(Drop-Down-Menü) vor.

1. Die Eingabe zur Selbstüberprüfung gibt einen Prozentsatz zurück, wie viel der Definition im eingegebenen Text enthalten ist

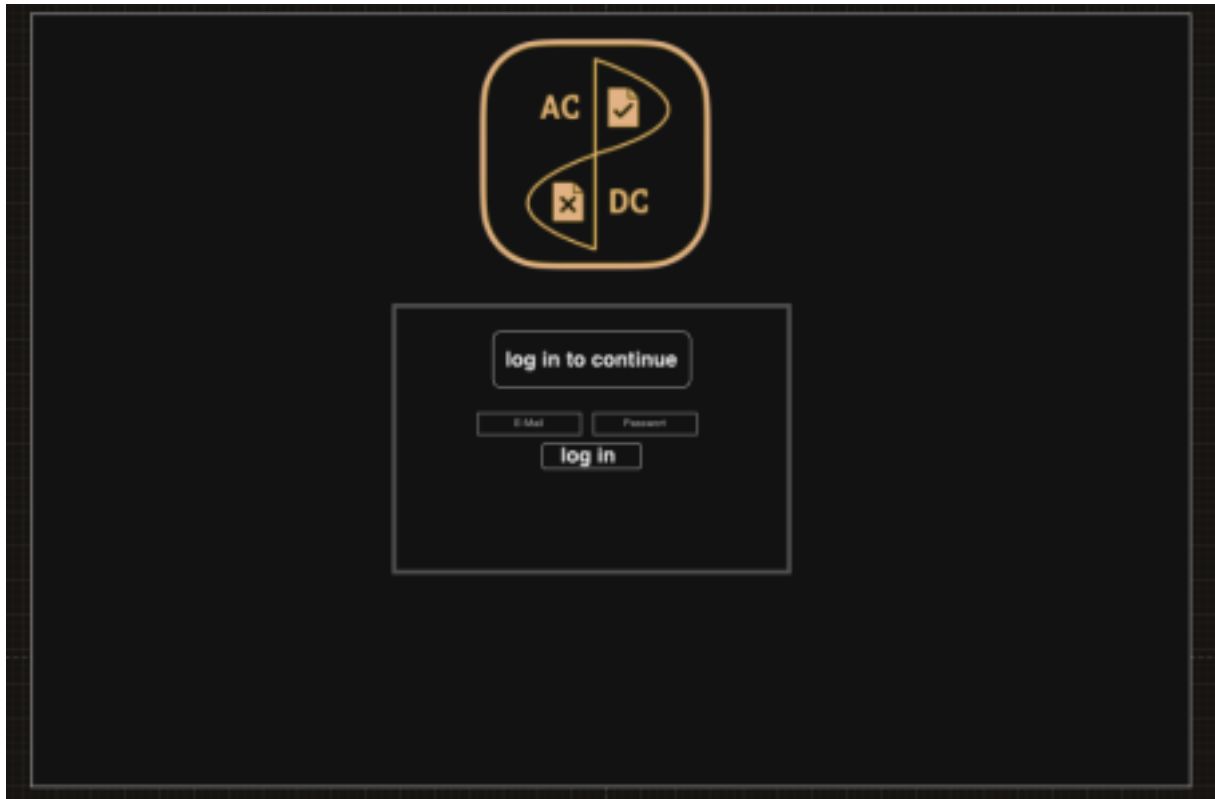
2. Enthält der Text mehr als 40% der Definition so kann die Karteikarte als erledigt markiert werden

3. Das Überspringen/Weiterblättern einer Karteikarte ist möglich

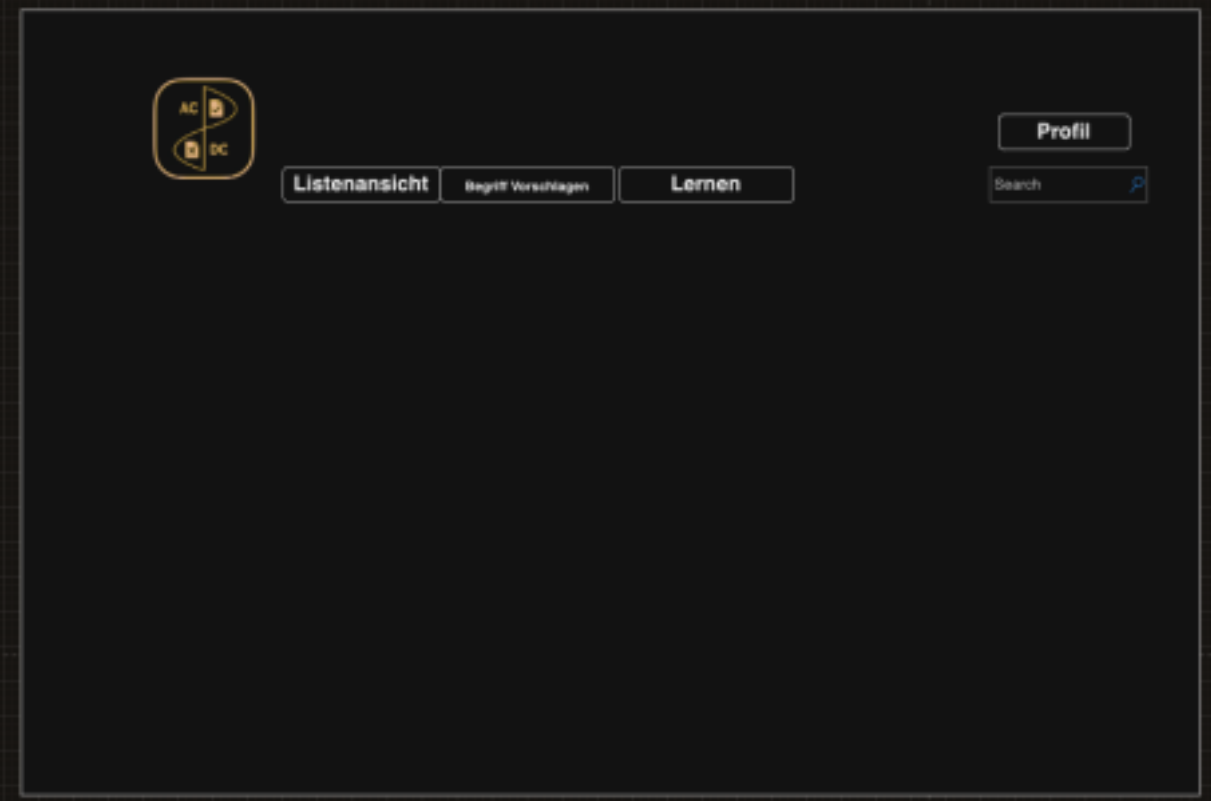
7. Die Begriffsansicht zeigt einen Begriff und die dazugehörige Definition an. Die Markierung der Lernfunktion ist erfolgreich und der markierte Begriff wird in der Lernfunktion abgefragt und zu dem jeweiligen Ordner hinzugefügt

8 Skizzen des Frontends

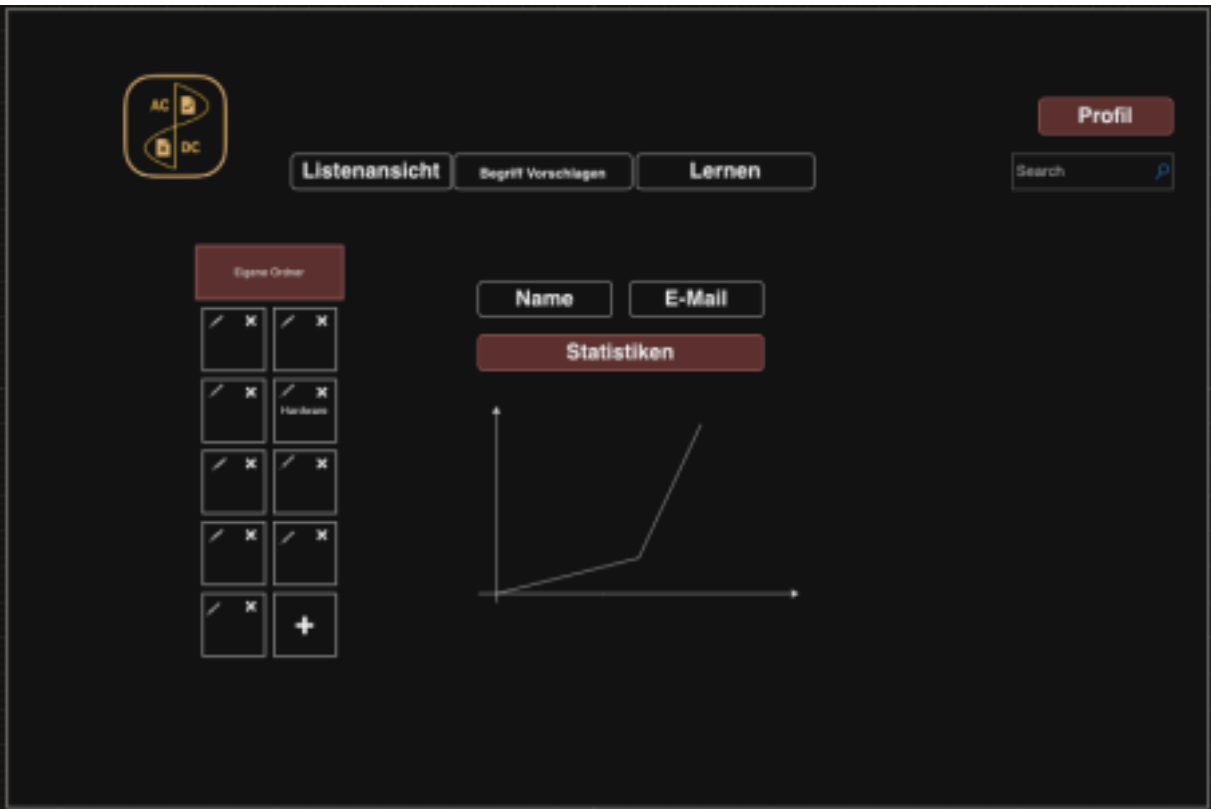
8.2 Abbildung 1 Login-Formular



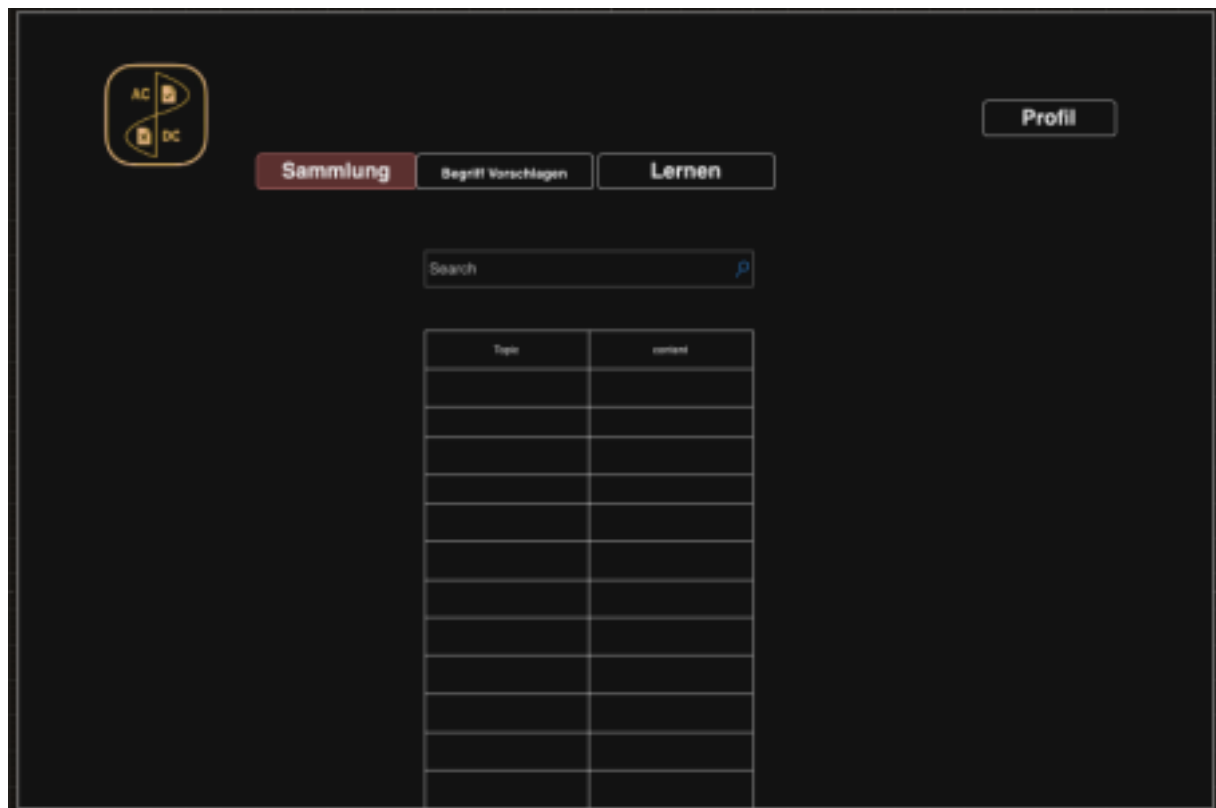
8.2 Abbildung 2 Benutzer-Dashboard



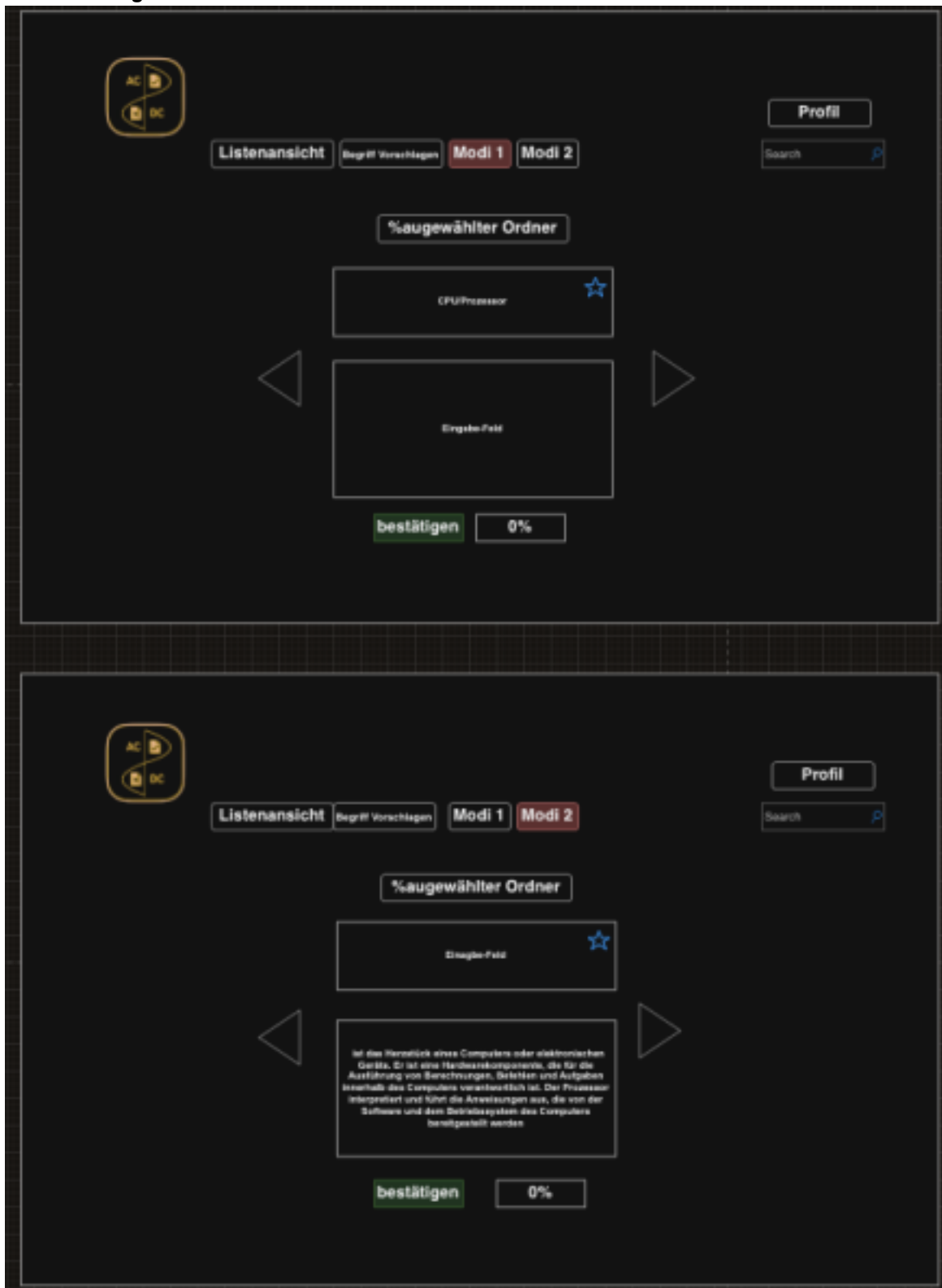
8.4 Abbildung 4 Benutzerprofil



8.5 Abbildung 5 Listenansicht



8.6 Abbildung 6 Lern-Ansicht



8.7 Abbildung 7 Begriffsansicht

