

# 1 Ausgangssituation und Zielsetzung

## 1.1 Ausgangssituation

Die BITLC GmbH bietet verkürzte Umschulungen zum Fachinformatiker in den Fachrichtungen **Anwendungsentwicklung** und **Systemintegration** an.

Im Rahmen der bisherigen IHK-Prüfungsvorbereitung wurden Defizite bei der Unterstützung der Teilnehmer festgestellt. Um eine effizientere und selbstständigere Vorbereitung zu ermöglichen, soll den Umschülern eine Softwarelösung zur Verfügung gestellt werden.

Bisher erfolgt die Prüfungsvorbereitung ausschließlich über analoge Materialien von Drittanbietern (z. B. Lernkarteikarten des u-Form Verlags). Die Geschäftsführung der BITLC GmbH erwartet durch die Einführung einer eigenen digitalen Lösung eine **Steigerung der Erfolgsquote** sowie eine **Verbesserung des Notendurchschnitts** der Absolventen.

## 1.2 Zielbestimmung

Ziel des Projekts ist die Entwicklung einer **webbasierten Anwendung** als Ergänzung zu den bestehenden Lernangeboten. Die Anwendung soll folgende Eigenschaften besitzen:

### Bereitstellung & Ausführung

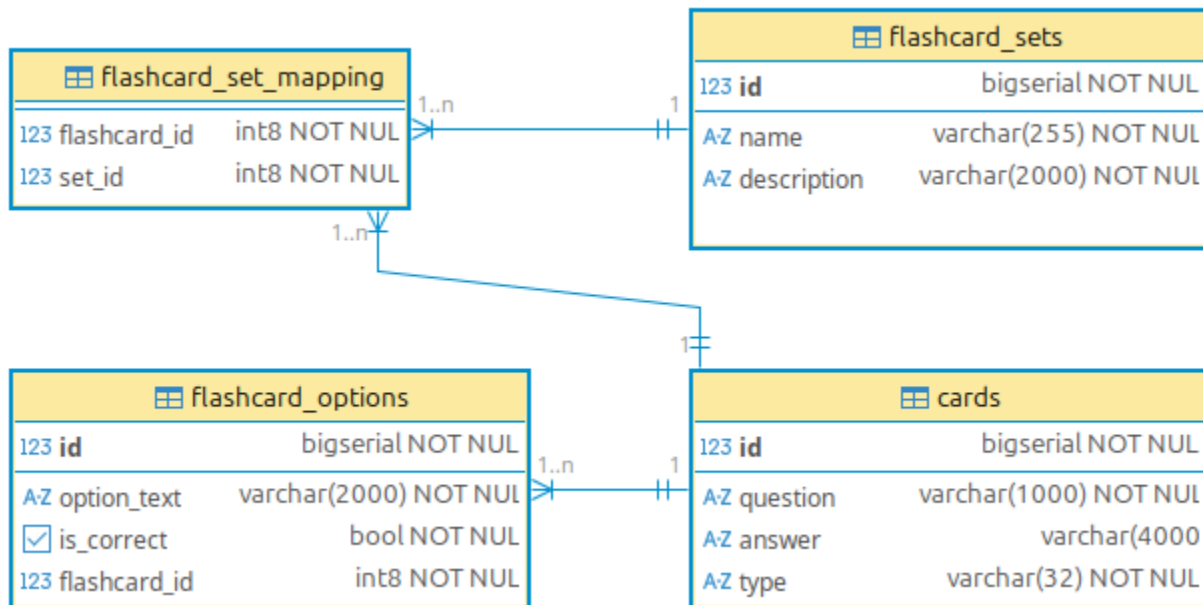
- Die Software wird als lokal installierbares Paket ausgeliefert.
- Zugriff erfolgt über einen aktuellen Webbrowser (Chrome, Firefox, Edge, Safari) auf Desktop- und Mobilgeräten.
- Deployment und Hosting der Anwendung erfolgen durch den Kunden selbst, wahlweise als öffentliche oder interne Unternehmenswebseite.

### Funktionalität

#### Kataloge & Inhalte

- Karteikarten werden in **Katalogen** organisiert (Titel, Beschreibung, optional Themen/Tags).
- Fachliche Inhalte (Kataloge/Fragen/Antworten) werden **serverseitig** in einer Datenbank gepflegt und über eine **Web-API** bereitgestellt.

## Datenmodell



## Fragetypen & Darstellung

- **Single-Choice (SC):** Genau **eine** Antwort ist korrekt. UI: Radio-Buttons.
- **Multiple-Choice (MC):** **Eine oder mehrere** Antworten können korrekt sein. Lösung gilt als **vollständig richtig**, wenn **alle** korrekten gewählt und **keine** falschen markiert wurden. UI: Checkboxes.
- **Lückentext (Fill-in):** Freitextfelder werden **case-insensitiv** verglichen; führende/trailing Leerzeichen werden entfernt; Unicode wird normalisiert. **Alternativlösungen** sind pro Feld optional als **RegEx** hinterlegbar. UI: Eingabefelder im Text.

## Katalogauswahl & Listen

- Startansicht mit **Katalogliste** (Titel, Kurzbeschreibung, Anzahl Fragen).
- Auswahl eines Katalogs führt zu einer **Fragenliste** (Listenansicht) und **Detailansicht** einzelner Fragen.

## Lernmodus (Üben)

- Fragen werden einzeln dargestellt; Nutzende können Antworten wählen bzw. eingeben.
- **Bewerten** zeigt **richtig/falsch** pro Frage; auf Wunsch **Lösung anzeigen** (inkl. optionalem Erklärungstext).
- **Fortschrittsanzeige** („Frage x/y“) und **Navigation** (Zurück/Weiter, zurück zur Liste/Start).

## Wireframes (Beispielhaft)

### Lernmodus

The wireframe shows a web browser window with the address bar displaying 'https://www.draw.io'. The page is titled 'Page 1'. The main content area is divided into three sections: a top 'HEADER', a middle 'Frage' (Question) section, and a bottom 'FOOTER'. The 'Frage' section is further divided into two columns. The left column contains 'Info' (Frage 2 / 30, Set, # Unbeantwortet, # Richtig beantwortet) and a 'Zurück' (Back) button. The right column contains the question text, a 'Vor' (Next) button, and a list of four settings (Setting 1, Setting 2, Setting 3, Setting 4) with radio buttons. Below the settings are two buttons: 'Antwort zeigen' (Show Answer) and 'Beantworten' (Answer).

### Lernkarte hinzufügen

The wireframe shows a web browser window with the address bar displaying 'https://www.draw.io'. The page is titled 'Page 1'. The main content area is divided into three sections: a top 'HEADER', a middle 'Frage' (Question) section, and a bottom 'FOOTER'. The 'Frage' section is further divided into two columns. The left column contains 'Info' (Frage 2 / 30, Set, Restzeit: 24:40, # Unbeantwortet) and a 'Zurück' (Back) button. The right column contains the question text, a 'Textarea' input field, a 'Fragentyp' (Question Type) dropdown menu (set to 'Multiple'), a 'Set' dropdown menu (set to 'none'), and a list of four text input fields (Text Input 1, Text Input 2, Text Input 3, Text Input 3). Below the text input fields are two buttons: 'Cancel' and 'Save'. A note on the right side of the page states: 'Für Textfragen wird eine andere passende komponente geladen' (For text questions, another suitable component is loaded).

## Prüfungssimulation

- Konfiguration: **Anzahl der Fragen** und **Zeitvorgabe** (Standard: alle Fragen des Katalogs, **1 Minute pro Frage**).
- Während der Simulation **keine Rückmeldung** zur Richtigkeit.
- **Ergebnisübersicht am Ende** (z. B. Anzahl richtig/falsch, Prozent). **Keine serverseitige Speicherung** der Ergebnisse.

## Bewertung & Fehlerbehandlung

- Einheitliche Bewertungslogik gemäß Fragetyp (SC/MC/Fill-in) wie oben definiert.
- Robuste **Fehlerbehandlung** bei API-Ausfällen (deutliche Meldung, Retry-Option), App bleibt bedienbar.

## Web-API (Abruf fachlicher Inhalte)

- Bereitstellung der Inhalte über GitHub Repository.
- Kernaufgabe: Kataloge und zugehörige Fragen abrufen; einzelne Fragen abrufen.

## Flashcard Controller Endpunkte

Endpunkt	Funktionalität
<b>GET</b> /api/flashcards?type={type}&set={setId}	Alle Karten optional gefiltert
<b>GET</b> /api/sets/{setId}/flashcards	Karten eines Sets
<b>GET</b> /api/flashcards/{flashcardId}	Karte nach ID
<b>POST</b> /api/flashcards	Neue Karte anlegen
<b>DELETE</b> /api/flashcards/{flashcardId}	Karte löschen

## Flashcard Set Controller Endpunkte

Endpunkt	Funktionalität
<b>GET</b> /api/sets	Alle Sets
<b>GET</b> /api/sets/{setId}	Set nach ID
<b>POST</b> /api/sets	Neues Set anlegen
<b>PUT</b> /api/sets	Set aktualisieren
<b>DELETE</b> /api/sets/{setId}	Set löschen

## 2 Produkteinsatz

### 2.1 Zielgruppen

Die Zielgruppen der Anwendung lassen sich in **primäre** und **sekundäre Nutzergruppen** unterteilen:

- **Primäre Zielgruppe:** Umschüler der Fachrichtungen *Anwendungsentwicklung* und *Systemintegration*, die sich im letzten Abschnitt ihrer Ausbildung befinden und auf die IHK-Prüfung (Teil I oder Teil II) vorbereiten.
- **Sekundäre Zielgruppe:** Dozenten der BITLC GmbH bzw. weiterer Bildungseinrichtungen. Diese übernehmen administrative Aufgaben, wie das Erstellen und Aktualisieren von Fragenkatalogen, sowie die Anpassung der Lerninhalte an aktuelle Prüfungsanforderungen.

### 2.2 Einsatzumgebung/Plattformen

- **Bereitstellung:**  
Die Anwendung wird als lokal nutzbares Softwarepaket ausgeliefert und kann auf typischen Serverumgebungen installiert und betrieben werden.
- **Plattformunabhängigkeit:**  
Das System ist auf allen gängigen Betriebssystemen (Windows, Linux, macOS) lauffähig.
- **Zugriff:**  
Der Zugriff erfolgt plattformübergreifend über moderne Webbrowser (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge).
- **Betriebsumgebung:**  
Das System ist für den kontinuierlichen Betrieb ausgelegt (Dauerbetrieb/24x7), sofern die serverseitige Infrastruktur dies unterstützt.

## 2.3 Sprachunterstützung

Die Anwendung ist auf den **deutschsprachigen Raum** ausgelegt.

- **Verkehrssprache:** Deutsch.
- **Internationalisierung:** Eine Mehrsprachigkeit wird nicht bereitgestellt.

## 2.4 Nutzungsszenarien

- **Eigenverantwortliches Lernen:** Lernende nutzen die Anwendung individuell zur Prüfungsvorbereitung. Sie können bestehende Fragenkataloge bearbeiten, nach Themengebieten lernen oder eigene Karteikarten ergänzen.
- **Bildungseinrichtungen:** Schulen und andere Institutionen stellen die Anwendung als Webanwendung zur Verfügung. Lehrende können die App konfigurieren, eigene Fragenkataloge erstellen und Lerninhalte institutionell bereitstellen.

# 3 Funktionale Anforderungen

Der Funktionsumfang der Software wird wie folgt beinhaltet:

## 3.1 Muss-Kriterien

- Plattformunabhängige Nutzung durch Umsetzung als Web-Applikation.
- Responsive Layout Anpassung für verschiedene Bildschirmgrößen.
- Web-API wird für den Fragekatalog bereitgestellt. Zugriff erfolgt auf ganze Kataloge und einzelne Fragen.
- Auswahlfunktion für einen Fragenkatalog sowie dessen einzelne Fragen in Listen- und Detailansicht.
- Navigationssystem mit Vor- und Zurück-Buttons zwischen einzelnen Fragen sowie Rückkehr zur Startseite und Katalogübersicht.
- System zur Anzeige von Lösungen und automatisches Bewertungssystem für die Beantwortung der Fragen.
- Enthaltene und unterstützte Fragetypen sind wie folgt angefertigt:
  1. **Single-Choice:** Genau eine valide Auswahl von mehreren Auswahlmöglichkeiten ausführbar.
  2. **Multiple-Choice:** Mindestens zwei valide Auswahl von mehreren Auswahlmöglichkeiten ausführbar.
  3. **Fill-in:** Die vom Benutzer eingegebene Antwort wird im Textfeld erfasst und im Hintergrund validiert.

## 3.2 Kann-Kriterien

- Zufallsreihenfolge von Fragen und Antworten als optionale Möglichkeit.
- Prüfungssimulation mit wählbarer Zeit und Anzahl von Fragen. Standardmäßig werden alle Fragen des Katalogs berücksichtigt und die Zeitvorgabe pro Frage beträgt eine Minute. Das Resultat der Prüfungssimulation wird ausschließlich am Ende als Übersicht angezeigt. Während der Simulation erfolgt keine Rückmeldung zur Richtigkeit der beantworteten Fragen.
- Die App soll dem Benutzer zunächst die Auswahl eines Themenbereichs ermöglichen und anschließend nur die zu diesem Bereich gehörenden Fragenkataloge anzeigen.

## 3.3 Abgrenzungskriterien

- Keine Authentifizierung innerhalb der Applikation.
- Keine Speicherung von Benutzerdaten.
- Keine Speicherung von Simulationsergebnissen.

# 4 Nicht-funktionale Anforderungen

## 4.1 Zugriff auf die App

- Die Applikation unterstützt gängige internetfähige Endgeräte (PC, Laptop, Tablet, Smartphone) und der Zugriff erfolgt über alle moderne Webbrowser.

## 4.2 Design und UX

- Die Interaktion und die Benutzerfreundlichkeit mit der Applikation ist einfach und angenehm durch intuitive Navigation und klare Struktur.
- Zeitgemäßes, modernes und responsives Design wird mithilfe von Angular Material UI Component Library realisiert.

## 4.3 Datenvolumen

- Die Anwendung soll 7 Fragenkataloge zum Thema "Einrichten eines IT-gestützten Arbeitsplatzes" für die IHK-Abschlussprüfung Teil 1 mit insgesamt 270 Fragen enthalten.
- Weitere Fragen und Fragenkataloge können individuell erstellt werden.

# 5 Lieferumfang

## 5.1 Zu übergebende Dokumente

Zu übergeben wird die Projektverlaufs-Dokumentation, ein Benutzerhandbuch für den Betrieb sowie eine Schnittstellenbeschreibung der entwickelten Web-API jeweils als PDF-Datei zusammen mit der Software.

## 5.2 Bereitstellung der Software

Die Software wird als GitHub-Repository übergeben.

**Hinweis:** Abweichung zum Lastenheft „ZIP + Cloud“.

Die Übergabe via Repository ermöglicht Versionierung und Nachvollziehbarkeit.

### Vorgaben:

- Das Repository enthält keinen Ordner node\_modules
- Alle für den Betrieb notwendigen Dateien sind enthalten
- Im Frontend wird Angular 19 benutzt
- Im Backend Spring Boot, Java 21, Docker sowie PostgreSQL
- Die Einrichtung von Docker und PostgreSQL ist in der README beschrieben
- Für die Übergabe wird Zugriff auf das GitHub-Repository (mind. Lese-Recht) bereitgestellt.

# 6 Abnahmekriterien

## 6.1 Qualitätsziele

- **Funktionalität:** Den Lernenden wird das volle Spektrum der vorhandenen Fragenkataloge zur Auswahl angeboten. Die von den Lernenden gegebenen Antworten werden anhand der in den Katalogen hinterlegten Antworten korrekt bewertet.
- **Zuverlässigkeit:** Die App kann im Routinebetrieb stabil genutzt werden. Fehler treten nur durch Ausfälle im Internet oder in der Stromversorgung auf.
- **Benutzbarkeit:** Die Bedienung wird robust, intuitiv, einfach und gut strukturiert gestaltet sein, damit die Lernenden das System ohne Schulungsaufwand nutzen können.
- **Effizienz:** Die Performance der Anwendung wird so ausgelegt sein, dass die Lernenden ohne längere Wartezeiten agieren können.
- **Änderbarkeit:** In der Programmstruktur wird die Erweiterbarkeit mit berücksichtigt, um später Änderungen leicht integrieren zu können.
- **Übertragbarkeit:** Die App kann auf andere Hardware- und Betriebssysteme mit Standardbrowsern übertragen werden.



## 6.2 Testszenarien

### T1 Browserzugriff (Bedienbarkeit der Anwendung)

**Ziel:** Grundlegende Erreichbarkeit und Bedienbarkeit der App sicherstellen.

**Vorbedingungen:** App ist bereitgestellt, aktueller Standardbrowser, Netzwerk verfügbar.

**Schritte:**

1. Startseite öffnen
2. Zu Katalog-, Fragen- und Detailansicht navigieren

**Erwartetes Ergebnis:** Alle Ansichten laden ohne Fehlermeldungen, UI-Elemente sind bedienbar, keine blockierenden Fehler in der Oberfläche.

**Bestanden-Kriterium:** Alle genannten Ansichten sind nutzbar, keine kritischen Fehler sichtbar.

### T2 Web-API-Anbindung (Datenabruf & Fehlerbehandlung)

**Ziel:** Korrekte Anzeige von Katalog- und Fragedaten sowie robuste Fehlerbehandlung.

**Vorbedingungen:** Backend & Datenbank laufen; mindestens ein Katalog mit Fragen vorhanden.

**Schritte:**

1. Katalogliste laden
2. Fragen eines Katalogs laden
3. Testweise API-Fehler provozieren (z. B. Dienst stoppen oder fehlerhaften Endpoint aufrufen)

**Erwartetes Ergebnis:** Bei Erfolg werden Titel, Anzahl und Inhalte korrekt angezeigt; bei Fehlern erscheint eine klare Fehlermeldung mit Hinweis (z. B. erneut versuchen).

**Bestanden-Kriterium:** Erfolgsfall liefert Daten korrekt, Fehlerfall zeigt verständliche Meldung, App bleibt bedienbar.

### T3 Reihenfolge der Fragen (Standard & Zufallsmodus)

**Ziel:** Verifizieren, dass die Standardreihenfolge stabil ist und der Zufallsmodus tatsächlich variiert.

**Vorbedingungen:** Ein Katalog mit mehreren Fragen, Umschalter für Zufallsmodus vorhanden.

Variante T3a Standardreihenfolge:

**Schritte:**

1. Zufallsmodus aus
2. Fragenliste laden
3. Liste aktualisieren/neuladen

**Erwartetes Ergebnis:** Reihenfolge ist stabil/definiert (z. B. sortiert nach ID), gleiche Menge an Fragen, keine Duplikate.

**Bestanden-Kriterium:** Reihenfolge identisch, Umfang identisch.

Variante T3b Zufallsmodus:

**Schritte:**

1. Zufallsmodus an
2. Fragenliste laden
3. Liste erneut laden

**Erwartetes Ergebnis:** Reihenfolge unterscheidet sich gegenüber vorherigem Lauf (mindestens eine Position geändert), gleiche Menge, keine Duplikate.

**Bestanden-Kriterium:** Reihenfolge variiert, Umfang identisch.

## T4 Lösung & Bewertung (Lernbereich)

**Ziel:** Korrekte Bewertung von Antworten und Anzeige der richtigen Lösung im Lernmodus.

**Vorbedingungen:** Katalog mit Single-/Multiple-Choice-Fragen sowie Freitext Fragen; „Lösung anzeigen“ im Lernmodus verfügbar.

**Schritte:**

1. Frage öffnen
2. Antwort(en) auswählen
3. Bewertung auslösen bzw. „Lösung anzeigen“

**Erwartetes Ergebnis:** Ergebnis „richtig/falsch“ ist korrekt, richtige Antwort(en) werden markiert/einblendbar, ggf. Erklärungstext sichtbar.

**Bestanden-Kriterium:** Bewertung stimmt mit Katalogdaten überein, Lösung wird korrekt angezeigt.

## T5 Navigation (Zustand & Datenintegrität)

**Ziel:** Konsistente Navigation ohne Datenverlust.

**Vorbedingungen:** App läuft, Frage im Detail geöffnet.

**Schritte:**

1. „Zurück/Weiter“ nutzen
2. Zur Fragenliste und zurück zum Detail wechseln
3. Zur Startseite und erneut in denselben Katalog navigieren

**Erwartetes Ergebnis:** Navigationspfade funktionieren, keine Inkonsistenzen, keine verlorenen Pflichtangaben/Zustände (sofern nicht bewusst verworfen).

**Bestanden-Kriterium:** Alle Navigationswege arbeiten erwartungsgemäß, kein Datenverlust, keine toten Links/Sackgassen.

# 7 Realisierung

## 7.1 Eingesetzte Sprachen, Frameworks und Tools

**Frontend:** Angular 19, TypeScript

**Backend:** Java 21, Spring Boot 3.x (REST, Data JPA, PostgreSQL)

**Datenbank:** PostgreSQL (z. B. 16)

**Container/Orchestrierung:** Docker, Docker Compose