

Anforderungsdokumentation – Praktikumsverwaltung

**Projekt im Rahmen der Umschulung zum
Fachinformatiker für Anwendungsentwicklung**

Autor: Emanuel Kronas

Datum: 29.09.2025

Inhaltsverzeichnis

1. Einleitung
2. Stakeholder
3. Systemkontext
4. Ziele & Vision
5. Funktionale Anforderungen
6. Nicht-funktionale Anforderungen
7. Randbedingungen
8. Use Cases
9. Datenmodell
10. Glossar
11. Qualitätskriterien
12. Risiken & Konflikte
13. Anforderungsmanagement
14. Offene Punkte

1. Einleitung

Das System „Praktikumsverwaltung“ dient der Erfassung, Verwaltung und Auswertung von Bewerbungen für Praktikumsstellen. Es ermöglicht autorisierten Benutzern die Verwaltung von Bewerbern sowie die Administration von Benutzerkonten. Das Dokument beschreibt die Anforderungen an dieses System.

Abgrenzung: Das System deckt nur die Verwaltung von Bewerbern und Benutzern ab. Es ist kein Bewerbungsportal für externe Kandidaten und bietet keine automatische Kommunikation.

2. Stakeholder

Im Projekt wirken verschiedene Stakeholder mit unterschiedlichen Rollen und Interessen:

Der Administrator ist verantwortlich für die Verwaltung der Benutzerkonten. Er legt neue Benutzer an, ändert ihre Daten oder löscht sie, wenn sie nicht mehr benötigt werden. Zusätzlich verwaltet er die Rollen und Rechte.

Der Sachbearbeiter bzw. Recruiter erfasst neue Bewerber, pflegt deren Daten und aktualisiert deren Bewerbungsstatus. Er ist der Hauptnutzer des Systems im Tagesgeschäft.

Die Ausbildungsleitung nutzt die Übersicht und die Exportfunktionen, um sich einen schnellen Überblick über den Stand der Bewerbungen zu verschaffen und Entscheidungen vorzubereiten.

Der Systembetreiber (IT) kümmert sich um die Installation, Wartung und den laufenden Betrieb des Systems.

3. Systemkontext

Fachlicher Kontext: Das System unterstützt die Verwaltung von Bewerbungen auf Praktikumsstellen. Es dient dazu, Bewerberdaten zentral zu erfassen und deren Bearbeitungsstatus transparent darzustellen.

Technischer Kontext: Das System ist als Webanwendung konzipiert. Der Zugriff erfolgt über einen Standard-Webbrowser. Die Daten werden in einer relationalen Datenbank gespeichert. Externe Schnittstellen zu anderen Systemen sind in dieser Version nicht vorgesehen.

4. Ziele & Vision

Das Hauptziel ist die effiziente und transparente Verwaltung von Bewerberdaten. Die Vision besteht darin, eine schlanke Lösung bereitzustellen, die schnell einsatzfähig ist und sich auf die Kernprozesse konzentriert.

5. Funktionale Anforderungen

Die wichtigsten funktionalen Anforderungen des Systems sind:

1. Benutzerverwaltung: Benutzer anlegen, bearbeiten und löschen; Rollen verwalten.
2. Authentifizierung: Anmeldung mit Benutzername und Passwort; Abmeldung (Logout).
3. Bewerberverwaltung: Bewerberdaten erfassen und bearbeiten; Status pflegen.
4. Dashboard: Übersicht über alle Bewerber nach Status.
5. Exportfunktion: Export der Bewerberdaten in CSV- oder PDF-Format.

6. Nicht-funktionale Anforderungen

Die wichtigsten nicht-funktionalen Anforderungen sind:

- Benutzerfreundlichkeit: Einfache, intuitive Bedienung.
- Sicherheit: Zugang nur über gesicherte Anmeldung.
- Performance: Antwortzeiten unter 2 Sekunden bei Standardaktionen.
- Portabilität: Lauffähigkeit auf jedem Webserver mit PHP ≥ 7.4 .

- Wartbarkeit: Klare Struktur und Trennung von Logik und Darstellung.

7. Randbedingungen

Technische Rahmenbedingungen für das Projekt:

- Programmiersprache: PHP
- Webserver: Apache oder Nginx
- Datenbank: MySQL oder vergleichbar
- Hosting: Lokaler oder interner Server

8. Use Cases

Die zentralen Anwendungsfälle des Systems sind:

- UC-1: Benutzer anlegen – Administrator legt neue Benutzer an.
- UC-2: Bewerber erfassen – Recruiter erfasst neue Bewerberdaten.
- UC-3: Bewerberstatus ändern – Recruiter aktualisiert den Status eines Bewerbers.
- UC-4: Exportfunktion nutzen – Ausbildungsleitung exportiert Bewerberdaten.
- UC-5: Login – Alle Benutzer melden sich mit Zugangsdaten an.

9. Datenmodell

Die wichtigsten Entitäten im Datenmodell sind:

- Benutzer: Attribute wie Benutzer-ID, Name, Benutzername, Passwort, Rolle-ID.
- Rolle: Attribute wie Rollen-ID, Rollenname (z. B. Administrator, Recruiter).
- Bewerber: Attribute wie Bewerber-ID, Name, E-Mail, Qualifikation, Status-ID.
- Status: Attribute wie Status-ID, Statusbezeichnung (neu, in Bearbeitung, angenommen, abgelehnt).

Beziehungen:

- Ein Benutzer hat genau eine Rolle.
- Ein Benutzer kann mehrere Bewerber erfassen.
- Jeder Bewerber hat genau einen Status.

11. Qualitätskriterien

Die Anforderungen sollen folgende Qualitätskriterien erfüllen:

- Korrektheit
- Vollständigkeit
- Eindeutigkeit
- Konsistenz
- Testbarkeit
- Nachvollziehbarkeit

12. Risiken & Konflikte

Mögliche Risiken und Konflikte sind:

- Zielkonflikt zwischen Datenschutzerfordernungen und einfacher Bedienbarkeit.
- Änderungen der Anforderungen während des Projekts erhöhen den Aufwand.
- Abhängigkeit von technischen Rahmenbedingungen (z. B. PHP-Version).

13. Anforderungsmanagement

Das Anforderungsmanagement umfasst:

- Versionierung aller Anforderungen
- Geregelte Änderungsprozesse mit Freigaben
- Eindeutige IDs für die Nachverfolgbarkeit

14. Offene Punkte

Noch offene Fragen für die weitere Umsetzung:

- Sollen Exportvorlagen (z. B. für Excel) anpassbar sein?
- Soll die Statusverwaltung individuell erweiterbar sein?
- Sollen automatische Statusänderungen (z. B. nach Fristablauf) implementiert werden?

10. Glossar

Administrator	Benutzer mit höchsten Rechten.
Bewerber	Person, die sich auf ein Praktikum bewirbt.
Status	Bearbeitungszustand einer Bewerbung (neu, in Bearbeitung, angenommen, abgelehnt).
Exportfunktion	Möglichkeit, Bewerberdaten als CSV oder PDF herunterzuladen.
Recruiter	Benutzer mit Rechten zur Bewerberverwaltung.
Dashboard	Übersichtsseite nach dem Login mit Bewerberdaten.
Benutzerverwaltung	Funktionen zur Anlage, Änderung und Löschung von Benutzerkonten.
Authentifizierung	Prozess zur Identifikation durch Benutzername/Passwort.
Datenexport	Export von Bewerberdaten in externe Dateien.
Statusverwaltung	Mechanismus zur Kennzeichnung von Bewerbungen mit Bearbeitungsstatus.