

Projekt: Anerkennungsbescheide-Assistenz für Stephan

Kontext

Wissenschaftlicher Mitarbeiter an Hochschule, bearbeitet ~75 Anerkennungsanträge/Semester. Studis wollen Prüfungsleistungen aus anderen Studiengängen anerkennen lassen.

Aktueller manueller Workflow

1. Antrag kommt im Funktionspostfach ein (~10 Module pro Antrag)
2. Stephan routet an zuständige Modulverantwortliche (nutzt Excel-Zuordnungstabelle)
3. Antworten von Profs einsammeln + nachfassen
4. Antworten zusammenfassen Bescheid schreiben (Word/PDF)
5. Review (Stephan schaut drüber)
6. Email an Studi

Was bereits existiert

TH Lübeck Prototyp (analysiert in /tmp/recog-ai-demo)

Tech Stack: Flask + ChromaDB (Vector-DB) + ChatGPT API (über OpenAI-kompatible Schnittstelle)

Workflow: 1. User: PDF Copy-Paste in Webform (max 10.000 Zeichen) 2. ChatGPT parst externes Modul JSON (Titel, Credits, Lernziele, Level, Prüfungsform) 3. ChromaDB: Semantic similarity search Top 5 interne Module 4. User wählt internes Modul ChatGPT vergleicht 1:1 5. Output: HTML side-by-side Vergleich + Empfehlung

Vergleichskriterien (gleichwertig, außer Arbeitsaufwand):

- Lernziele (primär): $\geq 80\%$ vollständig, $\geq 50\%$ teilweise, $< 50\%$ keine Anerkennung
- Credits: Externes \geq Internes OK; Internes » Externes max. teilweise
- Bildungsniveau: Bachelor vs. Master muss passen
- Prüfungsform: Nur wenn für beide vergleichbar
- Arbeitsaufwand: Nur wenn gut vergleichbar (sonst ignoriert)

Output: Side-by-side Modul-Vergleich (Prüfungsamt-Review-tauglich) + begründete Empfehlung

Was es NICHT löst:

- Keine finale Entscheidung (nur Empfehlung, Profs entscheiden)
- Kein Workflow-Management (Routing, Status-Tracking, Nachfassen)
- Kein rechtlich bindender Bescheid-Generator
- Keine Historisierung

Stephans Datengrundlage

- **Modulhandbuch als JSON** (bereits konvertiert/in Arbeit)
- **Excel-Zuordnungstabelle** (Ground Truth): Modul Modulverantwortlicher
 - Warum nötig: Modulhandbuch veraltet + Besonderheiten bei Anerkennungen
 - Wird in JSON übernommen oder als separater Lookup genutzt
- **Umrechnungsregeln** (in Excel)

Ziel: Assistenz-Workflow (NICHT Vollautomatisierung)

Support gewünscht:

- Routing-Vorschläge: "Modul X Prof Müller zuständig"

- Status-Dashboard: "Prof Schmidt antwortet seit 5 Tagen nicht"
- Bescheid-Entwurf basierend auf Prof-Antworten
- Konsistenz-Check: "Modul Y wurde letztes Semester schon anerkannt"

✗ NICHT gewünscht:

- Automatische Bescheide ohne Review
- Automatische Entscheidungen (bleiben bei Profs)
- Auto-Emails ohne Freigabe

Idealer Workflow: 1. Antrag → System schlägt Routing vor → Stephan bestätigt/korrigiert 2. System trackt Status → Dashboard zeigt offene Antworten 3. Antworten kommen → System generiert Bescheid-Entwurf → Stephan reviewed 4. Stephan gibt frei → System versendet

Constraints

- User: Kein Programmierer, Windows, M365 verfügbar
- Output: Professionelle rechtliche Dokumente (Briefkopf, sauberes Layout)
- ID-System für Anerkennungen innerhalb eines Antrags erforderlich
- Review-Step zwingend erforderlich (Stephan hat finales Wort)

Technische Komponenten

Setup-Phase (einmalig)

1. Stephans Modulhandbuch-JSON in Vector-DB laden
2. Excel-Zuordnung Modul → Verantwortlicher integrieren
3. Optional: Historische Anerkennungen importieren

Laufender Betrieb

- TH Lübeck System als Modul-Matching-Engine (oder eigene Instanz)
- Workflow-Layer für Koordination (zu definieren)
- Bescheid-Generator (Word/PDF Template)

Workflow-Vorschlag (von Jonathan)

1. Email von Studi → Tool erkennt automatisch passende Module, schlägt vor
2. Stephan wählt passendes Modul aus → Email an zuständigen Prof (Bitte um Prüfung)
3. Prof bewertet (manuell ODER mit Tool-Unterstützung) → antwortet Stephan
4. Tool generiert Bescheid (Zusage/Ablehnung) → Email an Studi

Offene Punkte:

- **Multi-Modul-Handling:** ~10 Module/Antrag → 10 separate Prof-Emails oder Batch? Übersicht bei 75 Anträge
× 10 = 750 Einzelprüfungen?
- **Status-Tracking:** Dashboard für "wer hat nicht geantwortet"? Nachfass-Reminder nach X Tagen?
- **Review vor Versand:** Automatischer Versand an Studi ODER Stephan gibt frei?
- **Prof-Zugang:** Brauchen Profs Login/Tool-Zugang oder Email-Antwort "Ja/Nein + Begründung" ausreichend?
- **ID-System:** Identifikation "Antrag #123, Modul 3/10" o.ä.?

Lösungsansätze (Evaluation ausstehend)

1. Power Automate + Word-Template - Uni-Infrastruktur, keine externen Tools
2. n8n/Make + Airtable + Documint - Flexibler, externes Tooling
3. Custom Flask App (basierend auf TH Prototyp) + Workflow-Extension

Offene Fragen

- Hat Stephan im JSON bereits Feld **verantwortlicher**, oder muss Excel-Lookup separat bleiben?
- Welche Email-Infrastruktur? (Outlook/Exchange via M365?)
- Wie sehen Anträge konkret aus? (Format, Struktur)
- Bestehende Word-Vorlagen für Bescheide?

Referenzen

- TH Lübeck Prototyp: <https://www.unidigital.news/th-luebeck-entwickelt-prototypen-fuer-ki-unterstuetzten-anerkennungsprozesse/>
- GitHub Prototype: <https://github.com/pascalhuerten/recog-ai-demo> (analysiert in /tmp/recog-ai-demo)