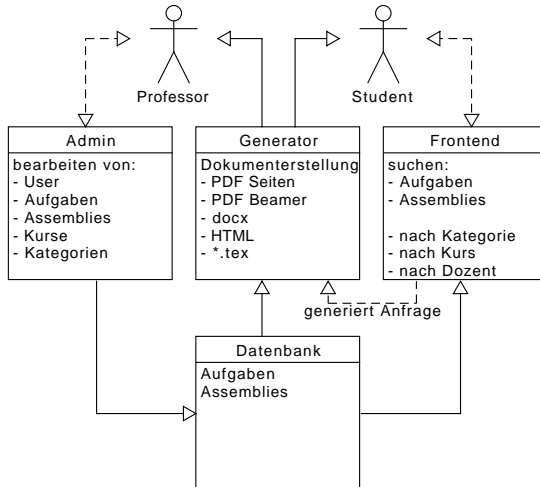


Dynamic Exercise Generation

Malte Koch / Jonathan Cyriax Brast / Pooya Aliyarzadeh
Frankfurt University of Applied Sciences
Faculty of Computer Science and Engineering

Wissen durch Praxis stärkt

Struktur



Django

- Python Webframework
- Beinhaltet viele Features die die Entwicklung einfacher machen
- Basiert auf dem DRY-Prinzip
- Explizite URLkonfiguration
- Bekannte Seiten¹: Mozilla, Spotify, Pinterest, Instagram

Django Vorteile

- Definition der Datenbank in Python
- Versionierbare Datenbankänderungen (Migrations)
- Datenbankobjekte und -queries können wie Pythonobjekte behandelt werden
- Automatische Generierung von Formularen aus Datenbankeinträgen

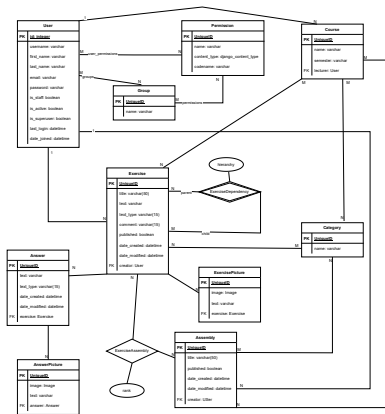
Django Vorteile

- Einfache Einbindung von Formularen in HTML
- HTML Vererbung
- IF-Bedingungen, For-Schleifen, Filter und weitere Features in HTML
- Anpassbare Admin Seite

Django Nachteile

- Kein einfacher Zugriff auf das HTML-Template der Adminseite
- Pythonkenntnisse nötig
- Auf den ersten Blick unübersichtlich/hohe Einstiegshürde

Datenbank ER-Diagramm



Pandoc

- Dokumentenkonverter
- Interne Baumrepräsentation von Dokumenten
- Unterstützt viele Formate
 - \leftrightarrow \LaTeX
 - \leftrightarrow Markdown
 - \leftrightarrow HTML
 - \leftrightarrow Microsoft Word docx
 - (\rightarrow Microsoft Power Point)
 - \rightarrow PDF

Pandoc

- keine Formeln in Powerpoint
- LaTeX Beamer macht *.tex Dateien statt Folien
- eigene Template-Sprache
- eigene Datenstrukturen

```
Pandoc(Meta({}), [Para([Str('sei'), Space(), Str('T'), Space(), Str('ein')...
```

- weniger Kontrolle über genaues Aussehen

Backend Endpoint

/generator?...

doctype=(pdf|docx|html|...)

&exercise=23&exercise=42

&assembly=42

&include_answers=1&title=0

- QueryDict wird an alle Funktionen weitergereicht
⇒ einfache Konfiguration / Feature-Toggles
- 400 BadRequest überall möglich (via Exceptions)
- Content-Type für PDF, docx, odt, Text
⇒ PDF & Text im Browser / Dokumente extern

Backend

