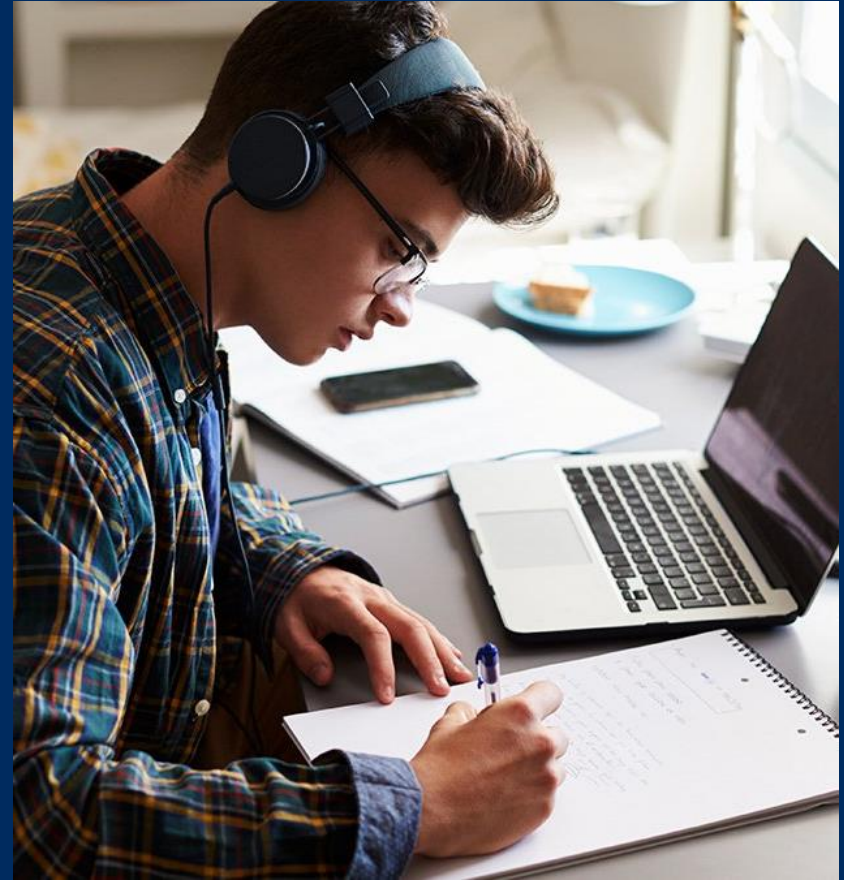


PrepPool – Aufgabenpool für Lehrpersonen

Annette Müller, Gisela Phillips, Melanie Rast, Patrick Weber

PrepPool

- Webapplikation mit Datenbank zur Verwaltung eines Pools von Übungs- und Prüfungsaufgaben
- Arbeitsblätter werden nach Thema, Bezeichnung und Schwierigkeitsniveau zusammengestellt und ausgegeben



User Stories – Überblick

Als LP möchte ich:

- Webseite PrepPool öffnen
- Account erstellen
- Zugriff mittels Login erhalten
- Neues Thema erfassen
- Neue Aufgabe erfassen
- Aufgaben auswählen
- Ausdruckbares Dokument erstellen
- Aufgabe löschen
- Aufgabe abändern
- Passwort anpassen
- Personenangaben anpassen
- Von PrepPool abmelden

Als Systemadmin möchte ich:

- Account an LP geben
- Account anpassen

User Stories – ein Beispiel

Als eine LP möchte ich nur mittels meines Logins Zugriff auf meine Aufgaben bekommen, um die Aufgaben vor unerlaubtem Zugriff zu schützen.

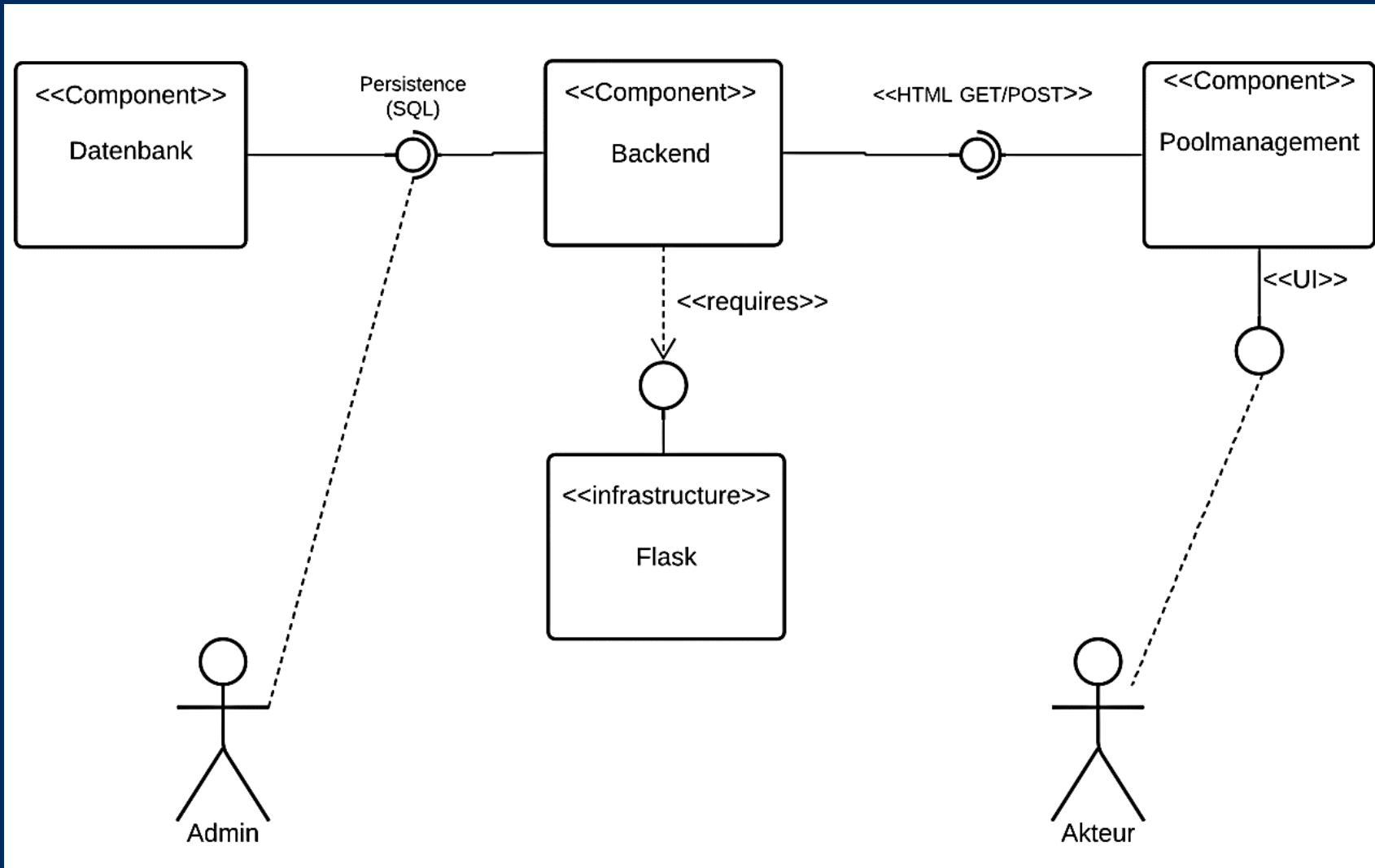
- **Definition Of Done:** Wenn der User die Email Adresse eines aktivierten Accounts mit dem richtigen Passwort eingibt, gelangt sie oder er in das Hauptmenu von PrepPool. Andernfalls erhält sie oder er eine Fehlermeldung, dass dieser Account nicht existiert.
- **Tasks:**
 - Erstellen eines HTML Formulars zur Eingabe von Email Adresse und Passwort
 - Erstellen einer Python Klasse zur Speicherung der Daten aus dem Formular
 - Implementieren der Funktionalität im Backend über eine Datenbank-Abfrage
 - Erstellen der HTML Seite mit der Fehlermeldung
- **Costs:** 1 Tag
- **Priority:** 1

User Stories – ein Beispiel

Als eine LP möchte ich ein neues Thema erfassen können, um meinen Aufgabenpool zu erweitern.

- **Notes/Limitations:** Die LP muss einen Account besitzen und eingeloggt sein.
- **Definition Of Done:** Nachdem die LP ein neues Thema eingegeben hat, wird dieses gespeichert. Nun ist es möglich, Aufgaben zu diesem Thema zu erfassen oder dieses Thema für die Erstellung eines Arbeitsblattes auszuwählen.
- **Tasks:**
 - Erstellen eines HTML Formulars zur Eingabe des Themas
 - Implementieren der Funktionalität im Backend über eine Datenbank-Abfrage
- **Costs:** 1 Tag
- **Priority:** 1

High-Level Design: Component Diagram

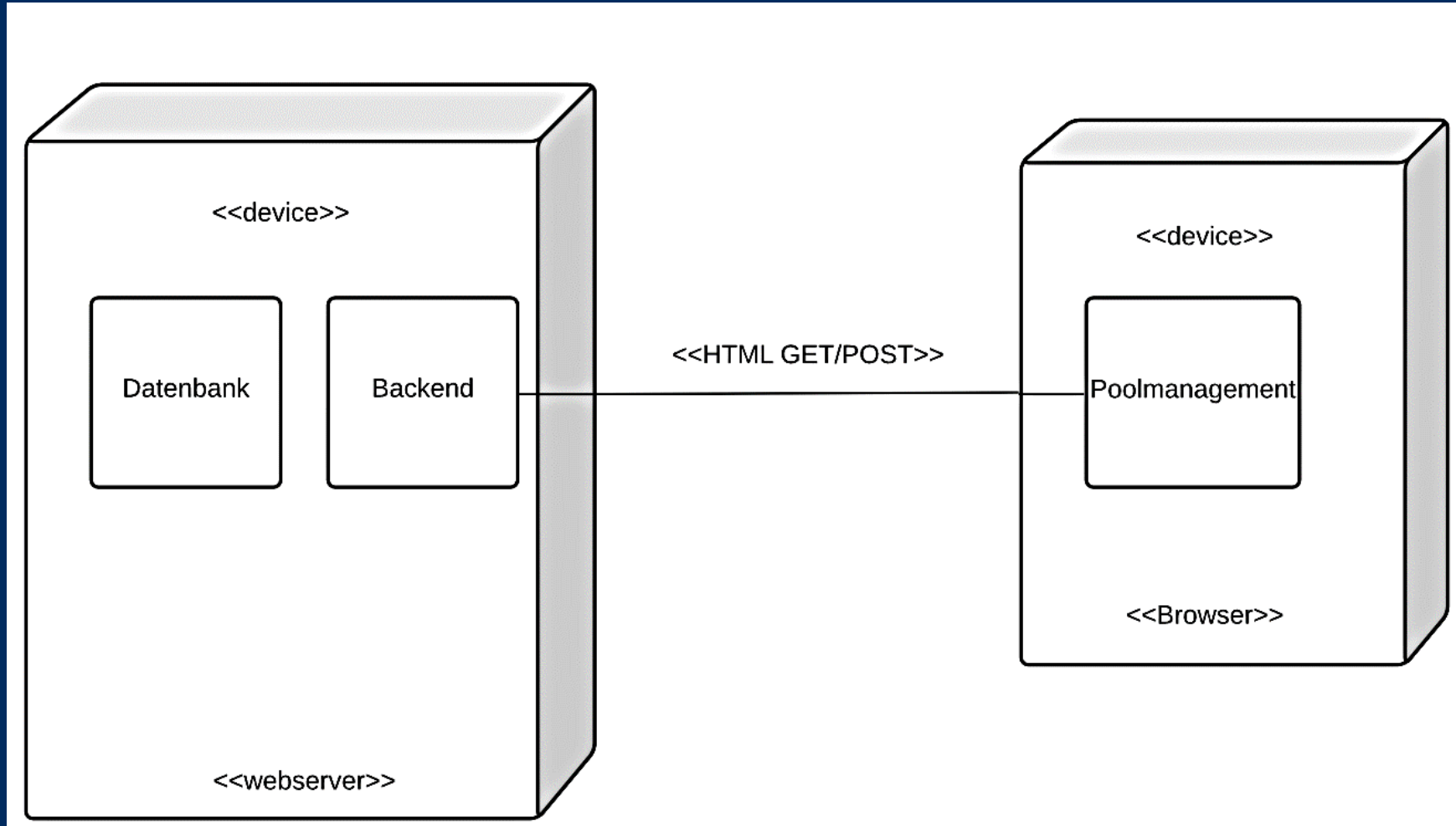


Was ist Flask?



- Micro Webframework (Python)
- Nutzt *Jinja2* (template engine) und *Werkzeug* (application library)
- Viele Entscheidungen bleiben dem Anwender überlassen

High-Level Design: Deployment Diagram



Sprint Plan: 1. Sprint

- #2 Account erstellen
- #3 Zugriff mittels Login
- #6 neues Thema erfassen
- #7 neue Aufgabe erfassen
- #16 Frontend-Zugriff