

Daten:

Benutzerverwaltung und Datenbank

md5

Benutzerscore | Passwort | Berechtigungsstufe \rightarrow standard ϕ

Enable \rightarrow um mit sozialen Netzwerk zu verbinden
od. Passwort zurücksetzen, validation

Benutzer registrieren: insert in die DB

Benutzer zurücksetzen: update der DB / email an user

Benutzer löschen: delete der DB

Berechtigung verwalten: update der DB

Einloggen: fetch auf die DB

π

Kritisch: online status? in der DB? Flag in der DB?

Ausloggen Session (beans)

Soziale Netzwerke einbinden via buttons und
eigenen Bereich für die jeweiligen Netzwerke

Kritisch: in DB gespeichert?

Achtung: Erstellung d. Benutzer im eigenen System
via Authentifizierung z.B. via profile ID

\Rightarrow An mongoDB für alles

Bücherverwaltung:

o Buch hochladen

→ pdf/epub mit Buch + Klappentext + Autor + Titel ... *konvertieren aller Bücher zu epub*
↳ Eingabeformular für User *continue via NextJS?*

▽ Feld zum hochladen von Dateien verfügbar? *Ja (PrimeFaces)*

▽ File extern gespeichert + Link darauf in DB? *externe Speicherung oder direkt in DB (Text)?*

▽ Falls externes File: eindeutige Dateinamen notwendig *hash als Dateiname*

o Buch herunterladen

→ Button für User

↳ ~~Speicherort~~ herunterladen im Browser

▽ Möglichkeiten zum upladen & downloaden *was* in JEE?

o Buch bearbeiten

→ Nur von User der geuploadet hat *(Zuordnung User & Buch beim Buch)*

→ Selbes Formular wie upload...? (vorhandene Daten automatisch in Felder einfügen) *Ja*

▽ Falls neue Datei hochgeladen, alte Datei löschen & Link erneuern
Bearbeiten des Buches Online, viel durch Mod, nur Bearbeitung ol. Metadaten,

o Buch melden

→ jeder angemeldete User über Button auf Detailsseite des Buches (+Text?)

→ Mails benachrichtigen *Bei Meldung Nachricht anzeigen dass Buch gemeldet*

o Buch löschen

→ nur User der geuploadet hat über Button

▽ alte Datei löschen!

Daten eingebracht werden:

- *Arch*

Then

- | | | | |
|-----------------------|----------------|--------|----------|
| - Beschreibung | Satznummer? | Ab SAT | MongoDB? |
| - Software | | | |
| - Anzahl Downloads | | | |
| - Bewertung | | | |
| - Kommentare | | | |
| - Autor | | | |
| - Uploader | → nicht ändern | | |
| erfolgreiche Funktion | | | |

Bild anzeigen:

- im Browser das Buch lesen können. **EPUB**
- Die derzeitige Seite speichern **T**
- Google Funktion **D**

Bibel sehen

- via Autor
- via Tikt
- via Reife Beurteilungen
- qualifiziert in 5 Stufen

Gib
Bewert

- scene system $\rightarrow \dots$

Bild kommentieren

- Kennenbuche der Benutzer ableiten
- Modulator-Steuerpflicht oft Kennenbuche
- Kennenbuche von Sozialrecht, Arbeitsanliegen

3. D. Johnson

Administratoren LF-40

INSY, SEW, DEZSYS

Benutzer verwalten LF-40-010

Administratoren verwalten registrierte Benutzer

CRUD-Prinzip

C... Create (Insert) von neuen Datensätzen importieren

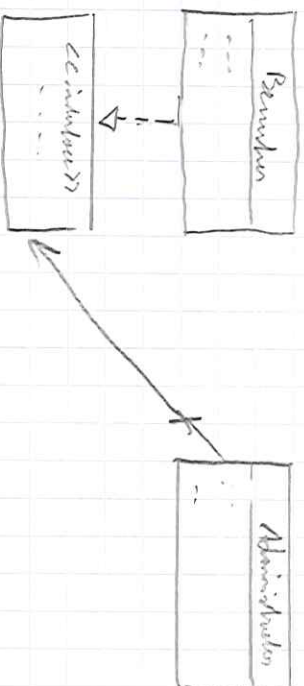
R... Read (Select) abfragen von registrierten Benutzern

U... Update (Update) bearbeiten, ändern, Bearbeitungen

D... Delete (Delete) löschen von Benutzern

- kleine Benutzer mit all ihren Attributen & Daten & neuen Attributen
(Document-basiert)

- kleine Administratoren verwalten Benutzer

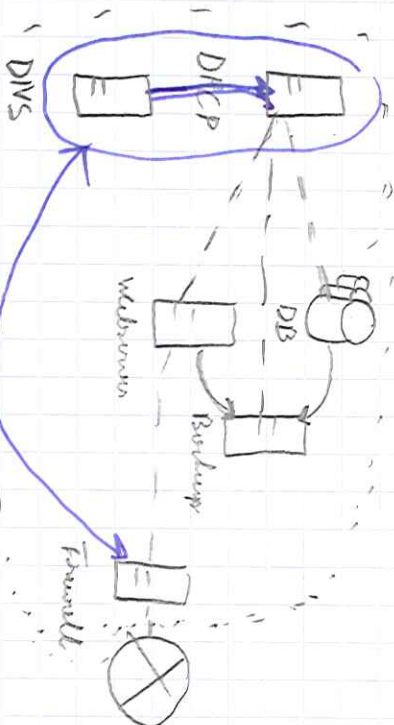


Benutzer erstellen LF-40-020

WTK, INSY

Benutzer-Sourcen bezieht Abfrage von DHCP (statische Adresse)

System Benutzer von Webbrowser (Radio) & DB-Sourcen (Datenbanken)



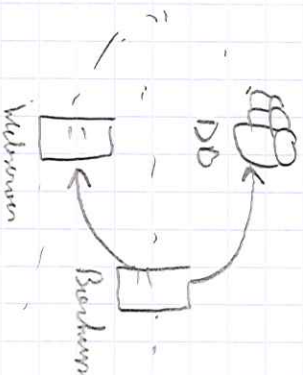
Buch-Prinzip

Benutzer = Document-basiert

Fehlertoleranz: Redundanz
(Code/DB)

- Backend einrichten LF-410-030 NWTk, ²
- aufbauen und LF-410-020
- Gesamt - oben Teil - Backend einrichten
- erzeugen Systemzustand einrichten

Pull-Prinzip



- Systemzustat einrichten LF-410-040 NWTk, DEFSYS
- Altsystemzustat (Spiceworks, Observeit, Nmap, Zabbix Core, Ganglia, OpenNMS, NetXMS, Zabbix)

= Verteilung auf Backend, auch dazu

- Punkt einrichten LF-410-050 NWTk
- aufbauen und LF-410-010
- Leistungsbildung konfigurieren
- Node konfigurieren / löschen
- Parameter System ändern



Nichtfunktionale Anforderungen

- Usability
 - einfache Föhrung, intuitives Verstöndnis d. Seite.
- Performance
 - Seitenladezeit < 1 Sekunde
 - Unblocken der Seite ohne Ladezeit
- Testföhrbarkeit
 - Loadbalance / mehrere Web/Da Server
- Fremdend-Testing
 - Testing mittels GUI Tester wie z.B. Selenium
- Nachwend-Testing
 - Softwaretests mit z.B. ~~TestDriven~~ Mockito
- Code Qualitöt
 - Prüfung durch Profiling
 - Codeabsicherung
- Dokumentieren
 - Sonaroc (erpariert)
- Sicherheit
 - Pw hashing
 - SSL Cert?
- Auto Install / Deployment / Security
 - mittels Vagrant etc