



**BookLab System**

(c) 2023 RENware Software Systems

# Arhitectura sistemului

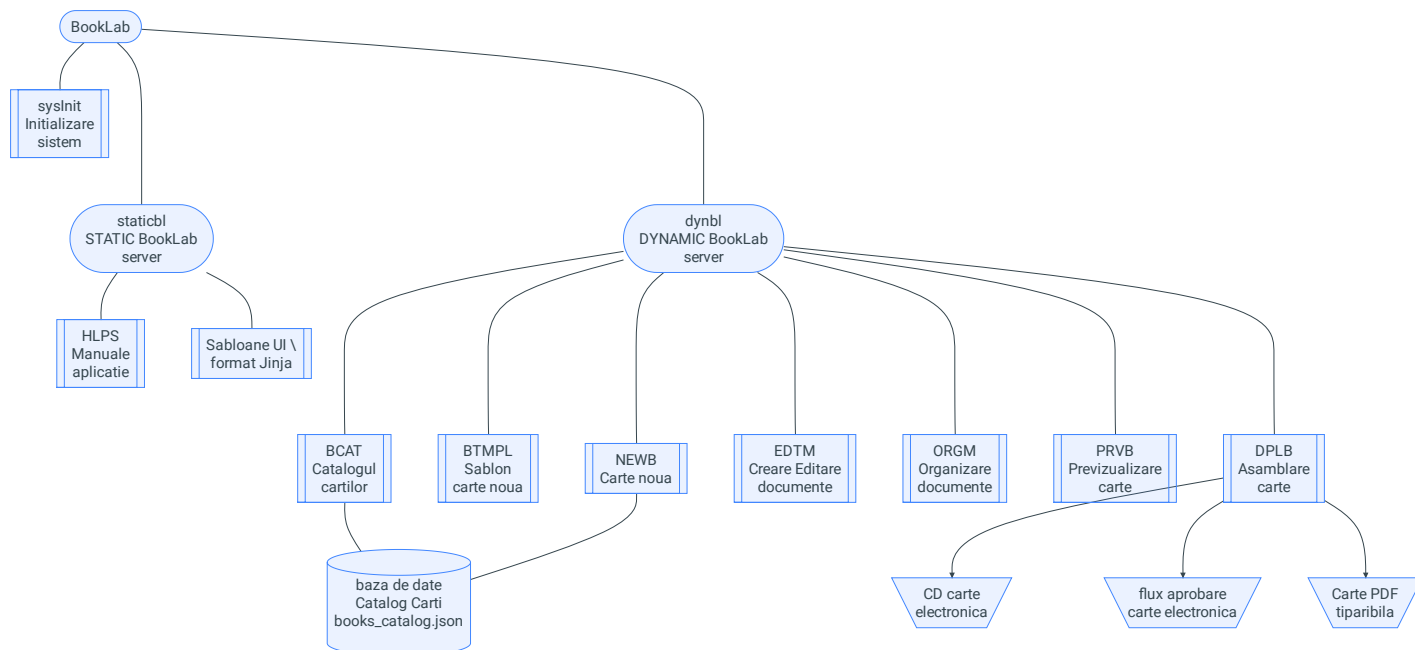
## **Cuprins:**

- [Arhitectura sistemului](#)
  - [Introducere](#)
  - [syslnit Initializare sistem](#)
  - [BCAT catalogul cartilor](#)
  - [BTMPL sablon carte noua](#)
  - [NEWB creare carte noua](#)
  - [EDTM preluare si editare continut](#)
  - [ORGM organizare structura carte](#)
  - [PRVB previzualizare si testare carte](#)
  - [DPLB asamblare publicare carte](#)
  - [HLPS help asistenta si manuale](#)
  - [Organizarea aplicatiei software](#)
  - [Note](#)

---

## Introducere

Arhitectura sistemului pleaca de la structura de baza a acestuia, structura redata "pe scurt" mai jos. Diagrama putin mai elaborata dar cu descrierea componentelor se gaseste in [documentul 130.02-Overview.md](#), [sectiunea "Structura si componenta BookLab"](#) ce se *recomanda a fi citit* inainte de a incepe studiul acestuia, pentru o mai buna intelegere a lui.



Referitor la *locatia fizica a componentelor* se face remarca ca toate componentele sistemului ce prezinta interfata vizuala cu utilizatorul au reprezentare atit in directorul `docs/` pentru partea de *front-end* (client side) cit si in radacina intregului sistem pentru partea de *back-end* (server-side). (Directorul `docs/` este generat prin compilarea aplicatiei in faza de development din directorul `doc_src/`.)

### ⚠ Conceptele de server si client

- Se atrage atentie asupra tratarii corecte din punct de vedere conceptual a notiunilor de `server` si `client`. Arhitectura conceptuala de client-server este pe deplin implementata cu respectarea tuturor principiilor aferente si o consintizare a acestui lucru va permite o intelegere corecta a detaliilor tehnice ale sistemului de catre echipa tehnica de lucru.
- In general in toata documentatia produsului prin termenul *CLIENT* se va intelege si partea de *CLIENT FRONTEND* iar prin cea de *SERVER* si partea de *SERVER BACKEND*.
- Locatiile pentru *partea de server static* (componenta `staticbl`) sunt situate dupa o organizare clasica de site static incepind din directorul `docs/` iar codul "de start" este prezent in fisierul `index.html` (presupunind ca la utilizarea unui HTTP server anume la dorinta clientului, acesta este configurat sa functioneze asa in lipsa specificarii unui fisier in rutele URL).
- In general (daca nu specifica in clar altfel) `code-name` al componentelor este acelasi atit pentru SERVER cit si pentru CLIENT. Diferentierea intre acestea se intimpla deoarece raspund servere diferite pentru porturile aferente (80 client si 4111 server). Astfel pe ruta "normala" a unei componente, de ex ptr *NEWB*: `http://booklab.../newb` va raspunde intii componenta CLIENT (prezenta sub forma unui `index.html` in directorul `docs/newb/`).

## sysInIt Initializare sistem

Pornirea sistemului comporta doua mari componente "on top", si anume:

- staticbl - STATIC BookLab** serverul HTTP static acesta va deservi partea de web compusa din "fisiere" statice. Acestea reprezinta fisierele HTML aferente interfetei UI dar si celelate fisiere "auxiliare" precum: CSS-uri, JS-uri,

imagini, etc

- **dynblsrv - DYNAMIC BookLab** serverul CGI dinamic ce permite rularea de scripturi Python server-side si deserveste componentele care executa operatiile necesare (de ex create carte noua). Acest server este cunoscut si sub numele de `BookLab API`

De asemenea referitor la initializarea sistemului trebuie spus si ca **prima instalare a acestuia** se face cu o serie de programe / scripturi. Nu se intra in detalii deoarece procedura de instalare a sistemului este una cit se poate de clasica pentru instalările executate manual pe sisteme Linux iar pentru sisteme Windows acesta este tot una clasica, cu un simplu wizard "standardizat" si care executa scrierea in toate locatiile necesare (ex Registry).

#### parametrii

- *code-name:* `sysInit`
- *locatie DYNAMIC BookLab:* `sysInit/`
- *locatie script PRIMA INSTALARE:* `setup/`
- *Document descriere detaliata:* [810.05a-sysInit\\_System\\_Process.md](#)

## BCAT catalogul cartilor

Acest modul este responsabil de catalogul (lista) cartilor si de sablonul de creare a unei carti. Astfel el contine urmatoarele elemente:

- **catalogul cartilor** ce este baza de date cu informatiile de baza despre carti. Acest catalog contine informatiile necesare pentru identificarea si referentierea unei carti in aplicatie. *Datele detaliate* aferente unei carti se gasesc in directorul aferent acesteia, director al carui nume se regaseste in acest catalog. Catalogul este stocat in format `JSON` si se gaseste in fisierul `books_catalog.json`.
- **interfata UI "Catalog carti"** ce este interfata CLIENT-UI<sup>1</sup> pentru managementul catalogului cartilor oferind operatii tip CRUD asupra bazei de date cu catalogul cartilor.

#### parametrii

- *code-name:* `bcat`
- *locatie DYNAMIC BookLab:* `docs/data/` cu baza de date catalogul cartilor `books_catalog.json`
- *locatie STATIC BookLab:* `docs/bcat/index.html` (`doc_src/index.md` before 1-st compilation)
- *Document descriere detaliata:* [810.05a-bcat\\_System\\_Process.md](#)

## BTMPL sablon carte noua

-#TODO...



Aceasta componenta refera **sablonul** implicit de creare a unei carti noi (si singurul in versiunea 1 a aplicatiei).  
Fisierele asociate acestui sablon se afla in subdirectorul `book_tmpl/`.

### parametrii

- *code-name*: `book_tmpl`
- *locatie DYNAMIC BookLab*: `docs/book_tmpl/`
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-btmpl\\_System\\_Process.md](#)

## NEWB creare carte noua

-#TODO...

**i** parametrii

- *code-name*: newb
- *locatie DYNAMIC BookLab*: newb/
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-newb\\_System\\_Process.md](#)

## EDTM preluare si editare continut

-#TODO...



### parametrii

- *code-name*: `edtm`
- *locatie DYNAMIC BookLab*: `edtm/`
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-edtm\\_System\\_Process.md](#)

## ORGM organizare structura carte

-#TODO...



### parametrii

- *code-name*: `orgm`
- *locatie DYNAMIC BookLab*: `orgm/`
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-orgm\\_System\\_Process.md](#)

## PRVB previzualizare si testare carte

-#TODO...



### parametrii

- *code-name*: prvb
- *port for running HTTP preview server*: 8111 (DIFFERENT than 80 which is reserved for main application)
- *locatie DYNAMIC BookLab*: prvb/
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-prvb\\_System\\_Process.md](#)



## DPLB asamblare publicare carte

-#TODO...



### parametrii

- *code-name*: `dplb`
- *locatie DYNAMIC BookLab*: `dplb/`
- *locatie STATIC BookLab*: ...nestabilit inca...
- *Document descriere detaliata*: [810.05a-dplb\\_System\\_Process.md](#)

## HLPS help asistenta si manuale

Manuale aplicatiei sunt de doua feluri (categorii) majore:

- **Manuale de utilizare**, acestea prezinta informatii pentru utilizarea curenta a aplicatiei. Catalogul acestora se poate accesa de la `<SERVER NAME>/help/880.30-EUMA_catalog.html`.
- **Manuale de configurare**, acestea prezinta informatii pentru configurarea aplicatiei. Catalogul acestora se poate accesa de la `<SERVER NAME>/help/880.30-ADMA_catalog.html`.

Pentru ambele tipuri de manuale se va prezenta un *catalog* ce contine lista tuturor manualelor disponibile in acea categorie. Lista contine numele manualului, o scurta descriere (daca numele nu este suficient de elocvent) si linkul catre manualul respectiv.

Manualele sunt localizate in sectiunea statica a aplicatiei (vezi si sectiunea "[Serverul HTTP static](#)") in directorul `docs/help/`.

## Organizarea aplicatiei software

Cod document metodologic: 810.05b Application Software Organization

In principiu aplicatia (sistemul) este organizata pe principiul *directoarelor de fisiere*. Aceasta permite o "rapida" regasire si recunoastere a informatiilor necesare chiar de la nivelul sistemului de operare (fara a necesita instrumente speciale ci doar comenzile existente in sistemul de operare). De asemenea o astfel de organizare este foarte apropiata de standardele clasice / originale ale aplicatiilor web.

Directoarele principale ale aplicatiei sunt enumerate la fiecare componenta din arhitectura si nu necesita explicatii suplimentare fata de cele scrise deja (vezi sectiunea [Arhitectura sistemului](#)). Detalii privind structura interna a fiecarui director aferent unei componente se regasesc in documentul de descriere detaliata aferent acelei componente. In situatii in care s-au considerat necesare diferite detalii pur specifice si "locale", in directorul componentei respective se gaseste un fisier `README_<cod-nume componenta>.md` ce prezinta aceste lucruri. Aceste documente se adreseaza in special dezvoltarilor aplicatiei (dar nu in mod obligatoriu) iar ca exemplu de ce poate contine un astfel de fisier este prezentarea modalitatii de pregatire a mediului local pentru dezvoltare.

## Note

- 
1. UI - desemneaza in general "Interfata Utilizator" (engl: User Interface) grafica, interactiva ce este afisata de catre browser (sau alta aplicatie GUI de acces) [↩](#)
- 

Last update: September 19, 2023