

Corso di Ingegneria del Software Deliverable di progetto	2024–2025
---	-----------

“Ingegneria del Software” 2024–2025

Docente: Prof. Angelo Furfaro

Gestore libreria personale

Data	<gg/mm/aaaa>
Documento	Documento Finale – D3

Team Members		
Nome e Cognome	Matricola	E-mail address
Leonardo Napoli	234364	npllrd02s30d086d@studenti.unical.it

Contents

	0.0.1	List of Challenging/Risky Requirements or Tasks	2
0.1		Stato dell'arte	2
	0.1.1	Obsidian	2
	0.1.2	Calibre	2
	0.1.3	Considerazioni	3

0.0.1 List of Challenging/Risky Requirements or Tasks

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
test	test	test	test

0.1 Stato dell'arte

A. Stato dell'Arte

Tra le applicazioni maggiormente utilizzate per la gestione di librerie ho potuto testare almeno 2 soluzioni utili in merito. Si tratta di Obsidian e Calibre.

0.1.1 Obsidian

Obsidian è un'applicazione più generalmente utilizzata per la creazione di note, mappe mentali e appunti personali, ma grazie alla grande possibilità che essa offre in termini di personalizzazione, si può adattare a qualsiasi utilizzo. Uno dei tanti utilizzi da me provati è stato proprio quello di mantenere la mia collezione di libri, con tanto di immagine di copertina, valutazione e commenti.

Vantaggi

- Personalizzazione praticamente illimitata grazie ai plugin open-source;
- Possibilità di creare collegamenti tra libri e/o commenti;
- Informazioni memorizzate in locale tramite file .md;

Svantaggi

- La gestione di una libreria devia dallo scopo generale dell'applicazione;
- Complessa inizializzazione da parte di un utente non tecnico;

0.1.2 Calibre

Calibre è un'applicazione open-source orientata alla gestione di Ebook, offre strumenti molto utili come la conversione di formati, gestione di metadati e sincronizzazione tra dispositivi.

Vantaggi

- Software maturo ed open-source;
- Estendibile con plugin;
- Totalmente in locale;

Svantaggi

- L'applicazione risulta inutile per la gestione di libri cartacei;
- Eccessive funzionalità offerte rendono l'interfaccia molto complessa.

0.1.3 Considerazioni

Osservando punti di forza e di debolezza delle applicazioni menzionate si è potuto derivare alcune funzionalità utili che saranno implementate nel sistema:

- Salvataggio dei dati in locale ed esportabile;
- Gestione semplice tramite interfaccia grafica;
- Funzioni limitate per un approccio minimale;
- Possibilità di aggiunta di funzionalità aggiuntive.

B. Raffinamento dei Requisiti

B.1 Servizi (con prioritizzazione)

1. **Memorizzazione in memoria secondaria.** Importanza alta, complessità bassa. Tutti i dati immagazzinati nel sistema dovranno essere trasferiti in memoria secondaria quando salvati.
2. **Creazione elementi.** Importanza alta, complessità bassa. Sarà possibile creare sia dei libri che delle collezioni di essi da gestire.
3. **Lettura elementi.** Importanza alta, complessità bassa. Sarà possibile accedere ai dati dei libri e delle collezioni memorizzate.
4. **Modifica elementi.** Importanza alta, complessità bassa. Sarà possibile modificare i dati relativi agli elementi memorizzati.
5. **Rimozione elementi.** Importanza alta, complessità bassa. Sarà possibile rimuovere elementi mantenendo il sistema in uno stato accettabile.
6. **Cambiamento stato libro.** Importanza alta, complessità bassa. Ogni libro avrà tra i suoi attributi lo stato di lettura, che potrà essere “letto”, “da leggere”, “in lettura”.
7. **Ricerca e ordinamento.** Importanza alta, complessità media. Ogni dato presente potrà essere ricercato e ordinato secondo i criteri disponibili.
8. **Virtualizzazione libreria.** Importanza media, complessità media. Il sistema formerà una libreria virtuale corrispondente a quella fisica, così da velocizzare la ricerca dei volumi.
9. **Interfaccia grafica.** Importanza alta, complessità alta. Il sistema con le sue funzionalità potrà essere interamente utilizzato tramite interfaccia grafica.

B.2 Requisiti non Funzionali

- **Multithreading.** Il sistema dovrà poter essere utilizzato in multithreading in maniera sicura.

- **Consistenza dei dati.** I dati inseriti dovranno rimanere consistenti a fronte di qualsiasi operazione messa a disposizione dal sistema.
- **Manutenibilità.** Il sistema deve essere ben strutturato così da garantire aggiornamenti ed aggiunta di funzionalità.
- **Portabilità.** Il sistema deve essere utilizzabile sugli OS più comuni, quindi Linux, MacOS e Windows.

B.3 Scenari d'uso dettagliati con relativi Use Case Diagrams

1. Aggiunta libro

- L'utente preme sul tasto "Aggiungi libro";
- L'utente inserisce le informazioni: titolo, autore, ISBN, genere;
- L'utente conferma le informazioni;
- L'utente preme sul tasto "Salva".

media/image1.png

2. Aggiunta collezione

- L'utente preme sul tasto "Aggiungi collezione";
- L'utente inserisce il nome della collezione;
- L'utente seleziona l'elemento desiderato;
- L'utente preme il tasto "Aggiungi";
- L'utente ripete le ultime 3 operazioni quante volte desidera;
- L'utente conferma le informazioni;
- L'utente preme sul tasto salva.

media/image2.png

3. Selezione elemento

- L'utente digita parola chiave nell'apposita casella;
- L'utente clicca su "Ricerca";
- Il sistema restituisce una lista di risultati;
- L'utente clicca sul risultato desiderato;

media/image3.png

4. Modifica dati

- L'utente seleziona l'elemento desiderato;
- L'utente preme sul tasto "Modifica";
- L'utente preme spunta la casella del campo che desidera modificare;
- L'utente digita il nuovo valore dell'attributo;
- L'utente preme su "Termina modifica";
- L'utente visualizza un'anteprima dell'elemento modificato;
- L'utente preme il tasto "Conferma".

media/image4.png

5. Elimina elemento

- L'utente ricerca l'elemento desiderato
- L'utente seleziona l'elemento desiderato;
-

A.4 Excluded Requirements

Descrivere i servizi eventualmente esclusi, e spiegare il perché

A.5 Assunzioni

<Briefly document, in this section, the most relevant requirement assumptions/decisions you had to made during your project>

C. Architettura Software

<IF RELEVANT, Report here both the static and the dynamic view of your system design, in terms of a Component Diagram, and their related Sequence Diagrams >

C.1 The static view of the system: Component Diagram

C.2 The dynamic view of the software architecture: Sequence Diagram

D. Dati e loro modellazione (se il sistema si interfaccia con un DBMS)

Definite le sorgenti di dati a voi necessarie per realizzare I servizi di cui sopra. Modellate tali dati tramite un ER o similari. Specificate se e quali di tali dati sono già forniti da applicativi esistenti.

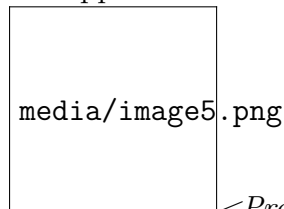
E. Scelte Progettuali (Design Decisions)

<Document here the **5** most important design decisions you had to take. You can use both a textual or a diagrammatic specification.>

F. Progettazione di Basso Livello

G. Spiegare come il progetto soddisfa i requisiti funzionali (FRs) e quelli non funzionali (NFRs) <Report in this section how the architectural and low level design you produced satisfies the FRs and the NFRs>

Appendix. Prototype



<Provide a brief report on your prototype, and especially: information on what you have implemented, how the implementation covers the FR and NFR, how the prototypes demonstrates your project correctness with respect to the FR and NFR. You may add some screenshots to describe what required above. Be ready to show your prototype during the oral examination>