Low Level Design Document

This Low Level Design (LLD) document outlines the core components and implementation details for **BookFinder - Library Search Interface**. The project provides a responsive frontend interface for users to search, filter, and view book details in a library catalog using React, TypeScript, HTML, CSS, and Tailwind.

---

1. System Components

Component	Description	Key Responsibilities
SearchBar	Input for search queries	Capture user input, trigger search
FilterPanel	Filter options (genre, author, etc.)	Update filter state, trigger search
BookList	Displays list of books	Render books, handle pagination
BookCard	Shows summary info for a book	Display book details, handle click
BookDetailModal	Modal for detailed book info	Show/hide modal, display details
App (Root)	Main app container	State management, layout

---

2. Class/Interface Overview

Name	Type	Relationships / Usage	Key Methods / Attributes
`IBook`	Interface	Used by BookList, BookCard, Modal	`id`, `title`, `author`, `genre`, `year`, `summary`
`IFilter`	Interface	Used by FilterPanel, App	`genre`, `author`, `year`
`App`	Component	Parent of all components	`state: books, filters, selectedBook`
`SearchBar`	Component	Child of App	`onSearch(query: string)`
`FilterPanel`	Component	Child of App	`onFilterChange(filter: IFilter)`
`BookList`	Component	Child of App	`books: IBook[]`
`BookCard`	Component	Child of BookList	`book: IBook`, `onClick()`
`BookDetailModal`	Component	Child of App	`book: IBook`, `onClose()`

---

3. Data Structure Overview

Model	Fields

`IBook`	`id: string`, `title: string`, `author: string`, `genre: string`, `year: number`, `summary: string`
`IFilter`	`genre?: string`, `author?: string`, `year?: number`

---

4. Algorithms / Logic

Search & Filter Flow (Pseudocode):

```
onSearchOrFilterChange(query, filters) {
  // Fetch or filter books based on query and filters
  const filteredBooks = allBooks.filter(book =>
    book.title.includes(query) &&
    (!filters.genre || book.genre === filters.genre) &&
    (!filters.author || book.author === filters.author) &&
    (!filters.year || book.year === filters.year)
  );
  setBooks(filteredBooks);
}
```

- Book selection opens `BookDetailModal` with selected book data.

---

5. Error Handling

Scenario	Handling Approach
No books found	Show “No results found” message
Network/API failure (if any)	Show error banner, allow retry
Invalid input (search/filter)	Validate input, show inline error
Modal open/close errors	Fallback to close modal, reset state

End of Document