

Section 1.1 – Project Overview

Project Description

BookVault is a web-based application designed to allow users to search for books, authors, and genres. The app retrieves book information based on user queries and presents matching results with detailed book and author descriptions. The primary goal is to provide a seamless and fast book discovery experience.

Project Management Style

We will follow a **Feature Driven Development (FDD)** approach, focusing on iterative development, regular feedback, and continuous improvements. Our development will consist of the following stages:

1. Develop an overall model
 - Perform an initial customer walk-through to gather requirements.
 - Identify key entities and define their relationships.
2. Build a features list
 - Identify all features required for the application.
 - Order features based on user needs and project goals.
3. Plan by feature
 - Categorize the list of features into classes.
 - Assign classes to an owning developer.
4. Design by feature
 - Develop detailed designs and sequence diagrams based on features.
5. Build by feature
 - Implement, test, and integrate features iteratively.
 - Conduct regular code reviews to determine necessary changes.
 - Promote finalized code to the current build.

Potential Users

Our app is developed for the following potential users:

- **Casual readers** looking for book recommendations and details.
- **Librarians and book enthusiasts** who need an efficient way to find books by author, title, or genre.

Section 1.2 – Preliminary Project Requirements

Feature List (FDD Format)

1. Search Engine

- a. Search for the book by its title
 - b. Search for the authors by their name
 - c. Apply filters for books (popularity, ratings, publication date)
2. Genre Search
 - a. Display a list of available genres
 - b. Display the number of books belonging to each genre
 - c. Display the list of books belonging to each genre
3. Book Detail Page
 - a. Display the metadata, descriptions, and reviews for a book
 - b. Display the recommendations related to a book
4. Author Detail Page
 - a. Display the details and description of the author
 - b. Display a list of books written by the author

CSCI Breakdown of User Stories

1. Search Engine Module

- **1.1 Search by Title**
 - As a user, I want to search for books by title so that I can quickly find the book I am looking for.
- **1.2 Search by Author**
 - As a user, I want to search for books by author so that I can explore books written by a specific author.
 - As a user, I want to see all books written by an author in one place for easy browsing.
- **1.3 Filters & Sorting**
 - As a user, I want to filter search results by popularity, ratings, and publication date to refine my search.
 - As a user, I want sorting options (e.g., newest first, highest rated) so that I can organize search results.

2. Genre List Module

- **2.1 Search by Genre**
 - As a user, I want to see a list of available genres so that I can explore books in my preferred category.
 - As a user, I want to view the number of books available in each genre to gauge the variety.
 - As a user, I want to click on a genre and see all the books under it so that I can find something to read.

3. Book Detail Module

- **3.1 Book Metadata**

- As a user, I want to view book details, including title, author, genre, and description, to learn more about each book.
- As a user, I want to see a book cover image to recognize the book visually.

4. Author Detail Module

- **4.1 Author Information**

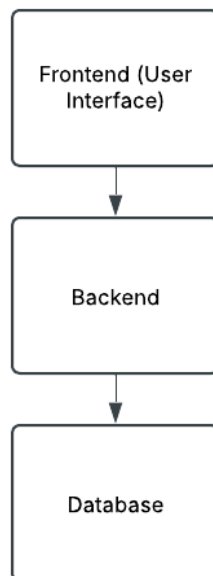
- As a user, I want to see an author's biography to learn about their background.
- As a user, I want to know key details such as awards, career highlights, and writing history.

- **4.2 List of Books by Author**

- As a user, I want to see all books written by an author so that I can explore more of their works.

Section 1.3 – Preliminary Project Design

System Architecture



1. Frontend:

- The user interface (UI) where the users interact with the app.

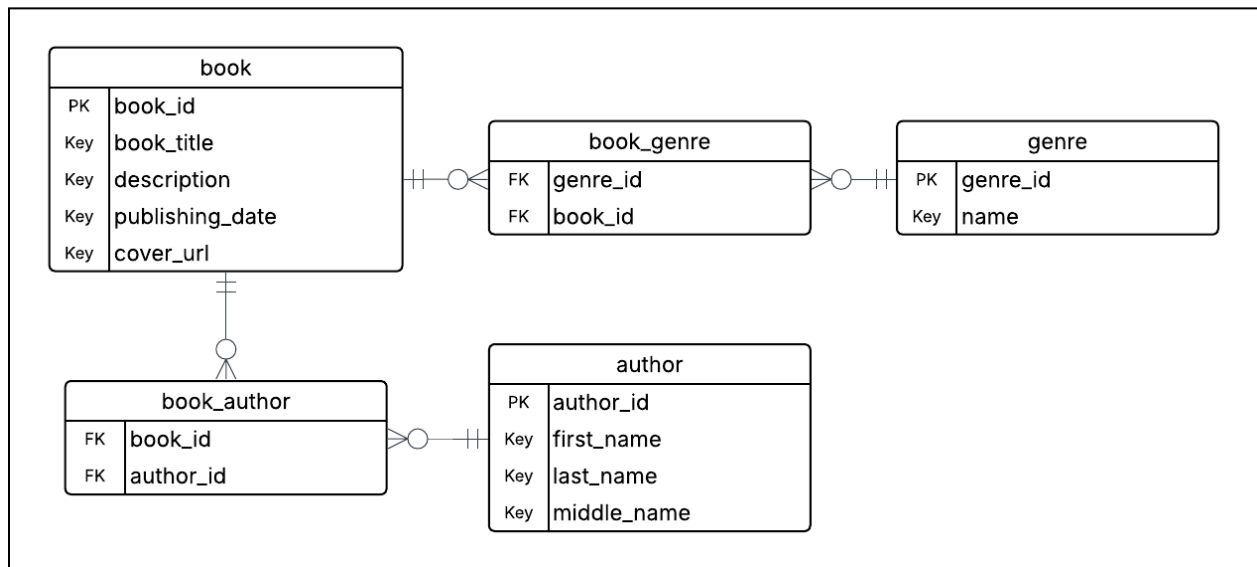
2. Backend

- Handles business logic, processes requests, and retrieves data.
- Built using a server-side framework.
- Connects to the database and external APIs.

3. Database

- Stores application data, such as book metadata, author details, and user preferences.
- Uses a relational database.

Entity Relationship Diagram



Relationships

1. Book - Genre (Many-to-Many)
 - a. A book can belong to many genres.
 - b. A genre can include many books.
2. Book - Author (Many-to-Many)
 - a. A book can be written by many authors.
 - b. An author can write many books.

Section 1.4 – Preliminary Development Schedule

Phase 1: Feature Planning & Modeling (Weeks 1-2)

Goal: Define the system model, identify features, and assign ownership.

1. Develop Overall Model:

- Create a high-level system architecture.

- Identify key entities and their relationships.

2. Build Feature List:

- Break down the project into features.
- Prioritize features based on user needs and project goals.
- Categorize features into classes.

3. Assign Feature Ownership:

- Assign each class of features to a developer for implementation.

Phase 2: Feature Design & Implementation (Weeks 3-8)

Goal: Design and implement the prioritized features iteratively.

1. Weeks 3-5: Implement Book Search Functionality

- Design and implement the search engine.
- Add filters and sorting options.
- Integrate with the database or external API for data retrieval.

2. Weeks 6-8: Develop Book Detail Pages & Metadata Integration

- Design and implement the book detail page.
- Add related recommendations and metadata.
- Ensure seamless integration with the search functionality.

Phase 3: Feature Testing & Refinement (Weeks 9-10)

Goal: Test features, gather feedback, and refine the application.

1. Unit and Integration Testing:

- Test individual features for functionality and performance.
- Perform integration testing to ensure features work together seamlessly.

2. Feature Review Meetings:

- Present features to potential customers for feedback.
- Identify areas for improvement or additional requirements.

3. Refinement:

- Refine features based on feedback.
- Fix bugs and optimize performance.

Section 1.5 – Development Tools

- Frontend
 - HTML/CSS
- Backend
 - Django
- Database and API
 - MySQL
 - Google Books API
 - Open Library API
- Other Tools:
 - Trello (Agile Project Management)
 - Slack (Team Communications)
 - Pycharm/Visual Studio Code (Code Editor)
 - Github (Version Control)
 - LucidChart (Diagrams)
 - GoogleDocs (Collaborative Documentation and Real-time Editing)