

Architecture Document

Feeler

1 INDEX

2	INTRODUCTION	3
2.1	REFERENCES	3
3	SYSTEM CONTEXT	4
4	CONTAINER- AND TECHNOLOGY CHOICE	5
5	COMPONENTS	6
6	CLASS DIAGRAM	7

2 INTRODUCTION

This document will define and document the application's software architecture from a high abstract- to a low concrete level.

All diagrams used in this document can be viewed in detail [here](#).

2.1 REFERENCES

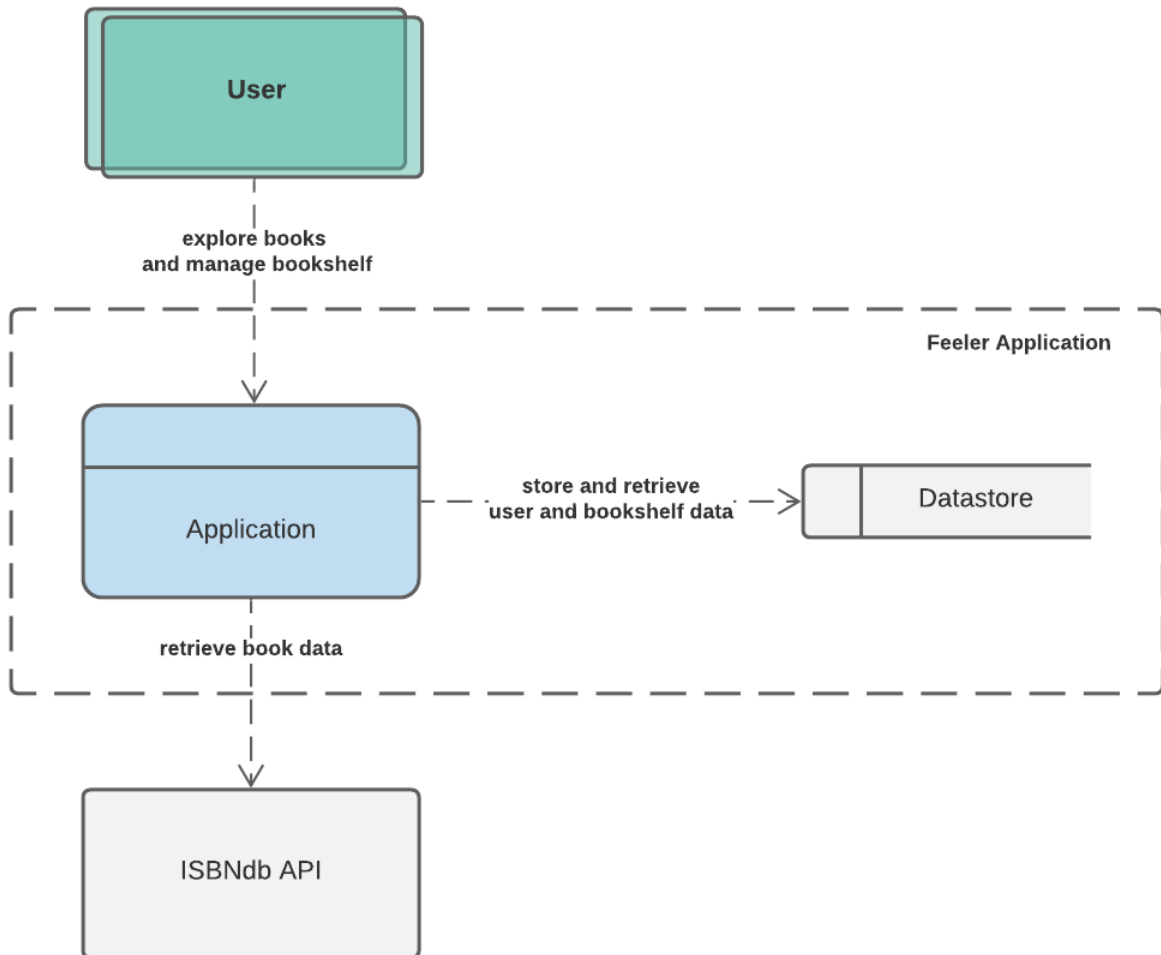
Where needed, this document will reference any external documents, but all documentation and code can be seen here: [documentation](#), [frontend code](#), [backend code](#).

3 SYSTEM CONTEXT

Here, the big picture of the application is defined.

The user uses the application to explore books and manage their bookshelf, the application stores and retrieves data from a datastore, and retrieves book data from the [ISBNdb API](#).

The following diagram can be found in-detail [here](#).

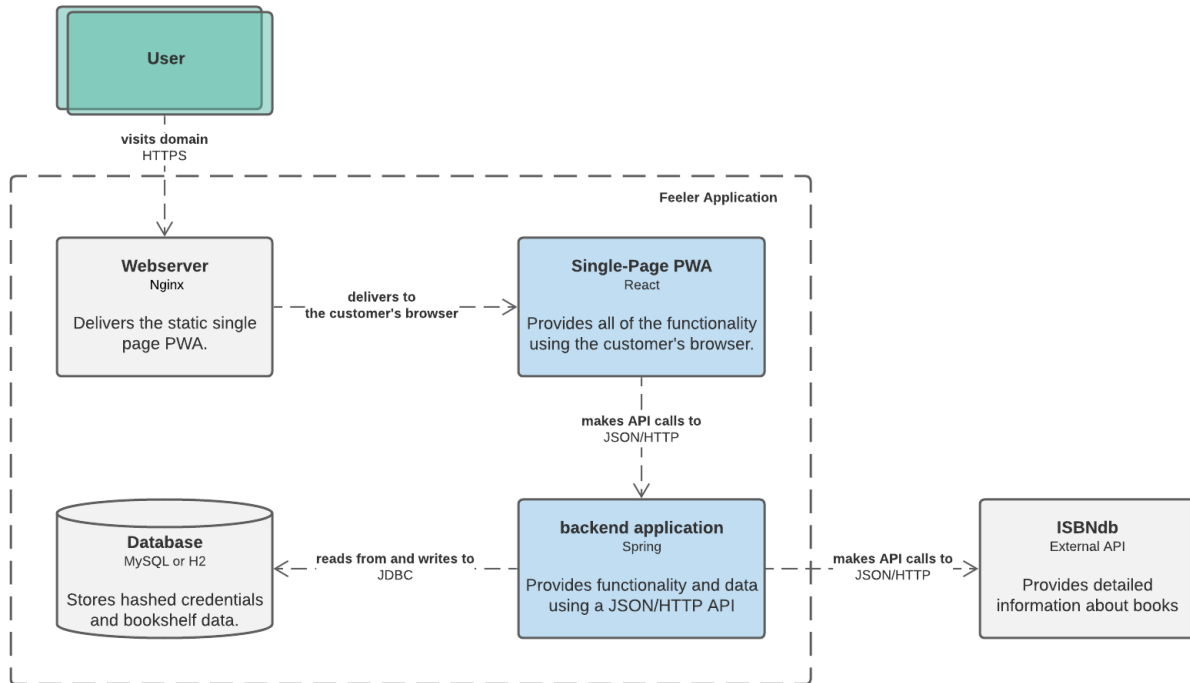


4 CONTAINER- AND TECHNOLOGY CHOICE

Here, the applications with their used technology are defined.

The Nginx webserver serves the user the single-page React progressive web application. The application makes HTTP calls to the Spring backend API, which gets its data from the MySQL database (or H2 for development), and the ISBNdb external API.

The following diagram can be found in-detail [here](#).

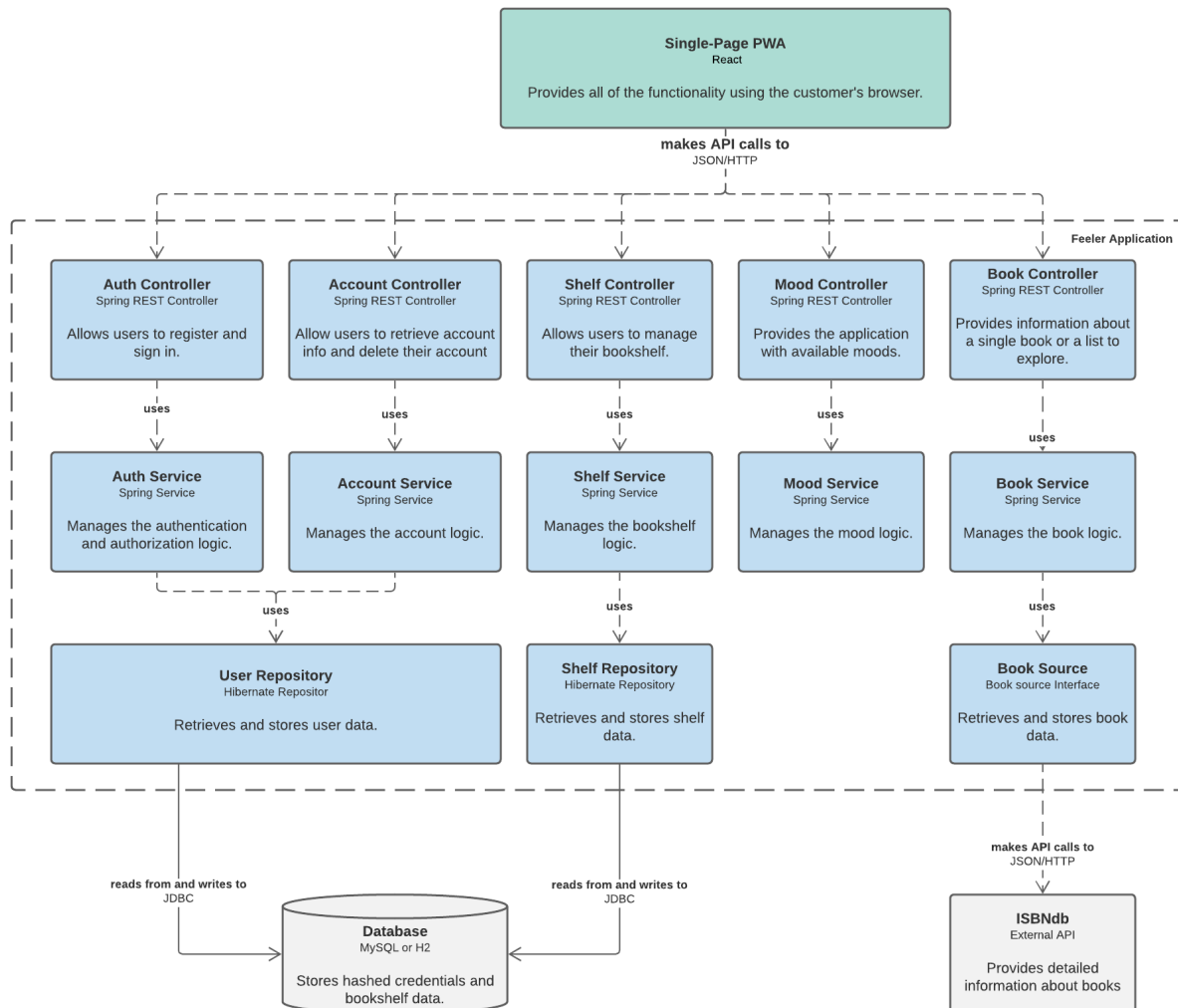


5 COMPONENTS

Here, the major structural building blocks with their interactions are defined.

The API call to the backend is received to the appropriate controller, which sends it up the chain to be handled by the associated service. The service handles the logic and can access the data sources using the data interface implementations.

The following diagram can be found in-detail [here](#).



6 CLASS DIAGRAM

Here the entire backend structure is defined.

The structure is composed of three layers: de API-controllers handling incoming API calls, the services handling the application's logic, and the data interfaces handling access to the appropriate data source.

Each layer has it's own objects: DTOs, models, and DAOs respectively. The mapper classes can be used to convert these objects.

The following diagram can be found in-detail [here](#).

