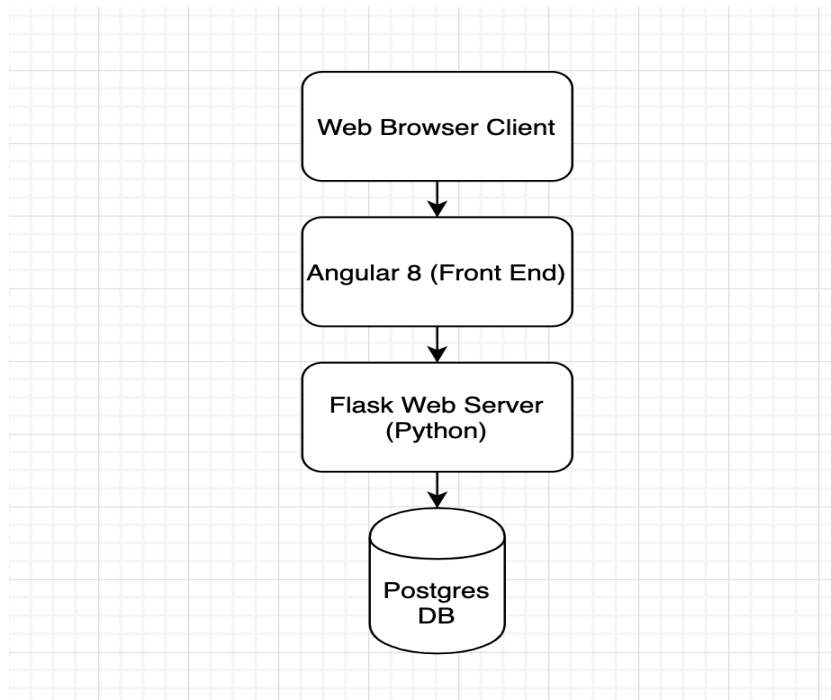<center>**DESIGN DOCUMENT**</center>

The **Bookish – Online Catalog** Library Management System is built as a Web Application using the three-tier architecture.

Diagram:

Front End:

Angular 8 + HTML5 + Typescript + SCSS + Bootstrap (Single Page Application)
- Used Angular's directives to iterate over the json file and display list of records in the HTML table
- Used bootstrap to provide styles for the entire application (Forms, Nav bar, tabs, tables)
- Created separate Angular components for each Tab sections
- Used Angular's reactive form to enable input validations and to provide helpful validation messages to the user.
- Used npm packet manager to install Angular application package dependencies
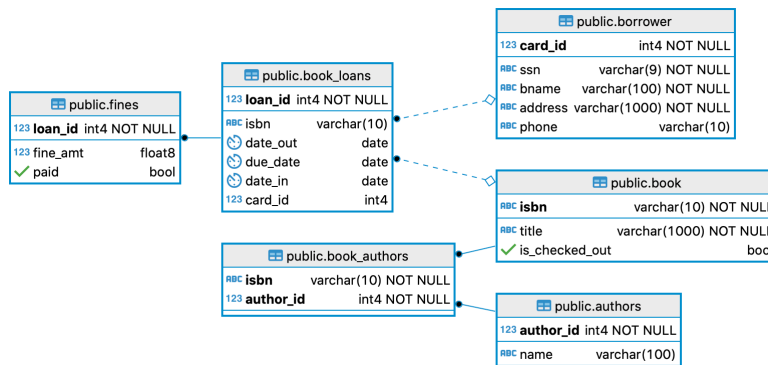
Server:

Python Flask framework
- Used psycopg2 adapter to interface with the Postgres Database using the database name and password

- Interfaces with the front end for obtaining data through stateless REST HTTP Verbs (GET, POST etc.)

Database:

Postgres: Connected using system credentials.

public.borrower

123 card_id	int4 NOT NULL
ABC ssn	varchar(9) NOT NULL
ABC bname	varchar(100) NOT NULL
ABC address	varchar(1000) NOT NULL
ABC phone	varchar(10)

public.book_loans

123 loan_id	int4 NOT NULL
ABC isbn	varchar(10)
date_out	date
due_date	date
date_in	date
123 card_id	int4

public.fines

123 loan_id	int4 NOT NULL
123 fine_amt	float8
✓ paid	bool

public.book

ABC isbn	varchar(10) NOT NULL
ABC title	varchar(1000) NOT NULL
✓ is_checked_out	bool

public.book_authors

ABC isbn	varchar(10) NOT NULL
123 author_id	int4 NOT NULL

public.authors

123 author_id	int4 NOT NULL
ABC name	varchar(100)

Design Decision and Assumptions:

- Search function is case insensitive and cannot distinguish between character accents
- This is a single user application.
- Implemented pagination for the results table when the list of records length exceeds 8
- Invalid messages are shown to the user in the form of red toaster message and valid messages are shown in the form of green toaster message.
- The application is split into 4 Tabs: Books, Check-in, Add borrower and fines
- It is assumed that the user operates this application in the same time zone always
- Card ID and Loan ID are auto increment fields
- Fines that are already paid are not displayed in the UI
- SSN and Phone numbers are considered pure numeric fields (No '- 's)
- SSN character limit – 9
- Phone number character limit – 10
- You can only check out 1 copy of book at a time
- You can check in multiple books at a time
- You can settle multiple fines at a time
- When navigating between tabs, selected tabs are activated while previously selected tabs are deactivated
- Is_checked_out column is added to the book table to keep track of availability status of that particular book
- Email address attribute is not fetched from Borrowers.csv
- Pages, Publisher and Cover attribute are not fetched from Books.csv