LetsBloom

Assignment

Note:

The instructions on how to run the application are provided in the GitHub README.md

Vanum Daniel Priyan

10th December, 2023

Contents:

1. Database Schema

a. Schema

The database has only one table, and it has the following characteristics:

- bookId: AutoField, Primary Key - An automatically incrementing unique identifier for each book.
- title: CharField (max_length=256), Not Null - The title of the book, limited to a maximum of 256 characters.
- author: CharField (max_length=64), Not Null - The author of the book, limited to a maximum of 64 characters.
- edition: IntegerField, Not Null - The edition number of the book.
- publisher: CharField (max_length=256), Not Null - The publisher of the book, limited to a maximum of 256 characters.
- language: CharField (max_length=32), Not Null - The language in which the book is written, limited to a maximum of 32 characters.
- publicationDate: DateField, Not Null - The publication date of the book.
- numberOfCopies: PositiveBigIntegerField, Not Null - The number of copies available for the book.

b. Candidate Keys

- "bookId" is a candidate key as well, serving as the primary key for the table. It provides a unique identifier for each book in the database.

- The combination of fields (title, author, edition, language, publisher) is specified as a composite unique constraint, making it a candidate key. This ensures that no two books in the database can have the same values for these fields.

c. Primary Key

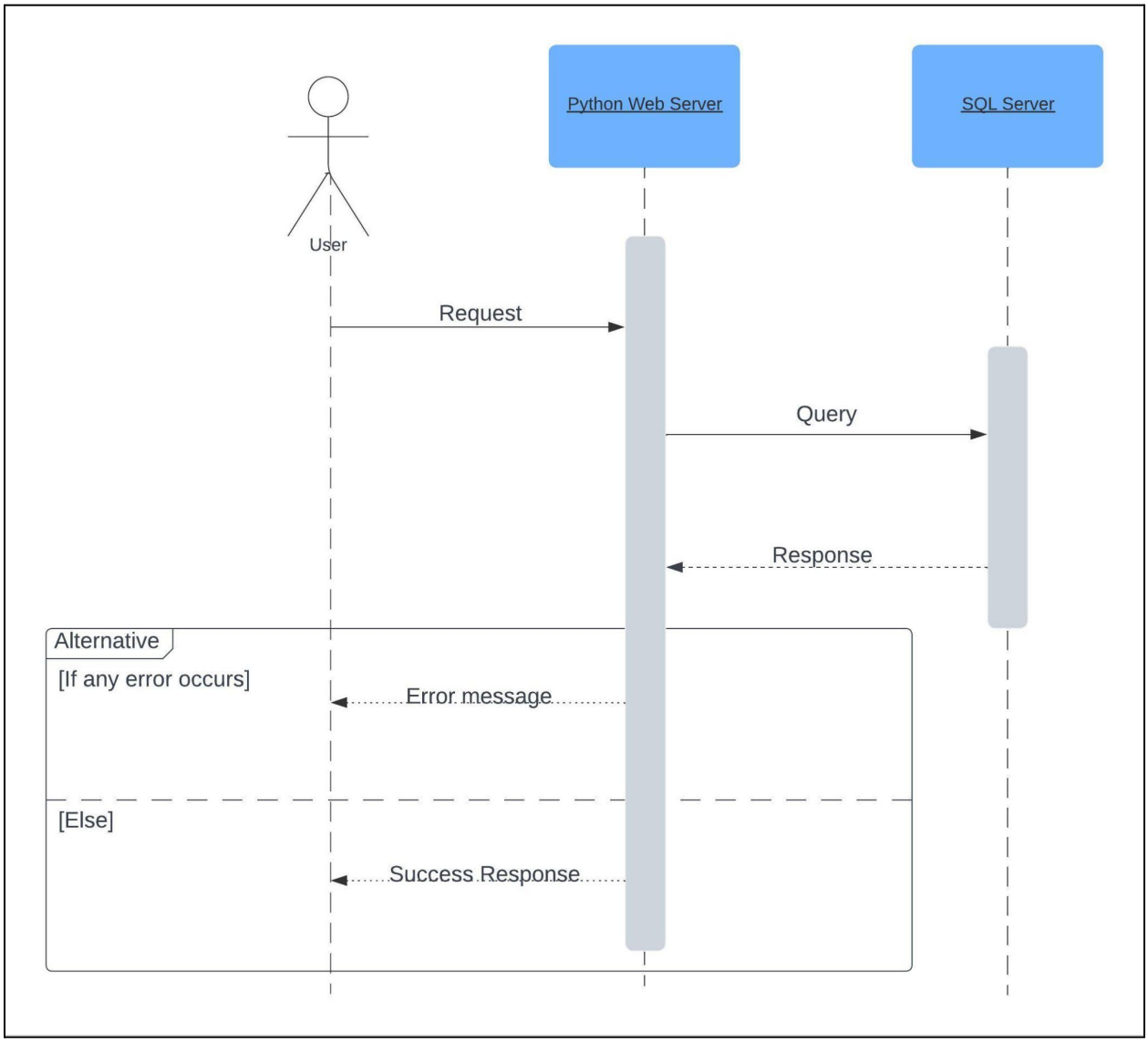
bookId is a candidate key as well, serving as the primary key for the table. It provides a unique identifier for each book in the database.

d. Details About Normal Form of the database

- The given schema is in 1NF as the attributes are atomic in nature 2NF, 3NF and BCNF
- Candidate keys - bookId and {title, author, edition, language, publisher}
- Non Key Attributes - publicationDate and numberOfCopies
    - As there is no Functional Dependency In the form X->Y where X is a proper subset of some Candidate Key and Y is a non-key Attribute. Hence the schema is in 2NF.
    - As there is no Functional Dependency In the form X->Y where X and Y are a set of non-key Attributes. Hence the schema is in 3NF.
    - As there is no Functional Dependency In the form X->Y where X and Y are prime Attributes. Hence the schema is in BCNF.

Hence the schema in Boyce-Codd Normal Form (BCNF) which eliminates various write anomalies.

2. Sequence Diagram



3. Endpoints

The application has 3 endpoints, and details of each endpoint are given below.

a. Endpoint 1: Retrieve All Books

- Endpoint URL: GET /api/books
- Description: Retrieve a list of all books available in the database.
- Request Parameters: None
- Request Query: None
- Request body: None
- Response: Can be of types
  - 200 OK: List of books is present in the response.
  - 500 Internal Server Error: Occurs when the server encounters an internal error or a database error.
- Notes
  - The response is a JSON array containing details of each book.
  - If there is a database connection issue or unknown error, a 500 response is received with appropriate response.
- Examples:

1. Success Response (200 OK)

- Response Body:

```
{
    "status": "success",
    "info": "Books fetched successfully",
    "data": [
        {
            "bookId": 1,
            "title": "1984",
            "author": "J.K. Rowling",
            "edition": 6,
            "publisher": "Wiley",
            "language": "Farsi (Persian)",
```

```
            "publicationDate": "2022-01-15",
            "numberOfCopies": 13
        },
        {
            "bookId": 2,
            "title": "To Kill a Mockingbird",
            "author": "Herman Melville",
            "edition": 9,
            "publisher": "Elsevier",
            "language": "German",
            "publicationDate": "2022-04-04",
            "numberOfCopies": 17
        },
        // more books...
    ]
}
```

2. Error Response

- HTTP Status: 500 Internal Server Error
- Response Body:
    - For database connection error

```
{
  "status": "error",
  "info": "Database connection error"
}
```

    - Unexpected errors

```
{
  "status": "error",
  "info": "An unknown error occurred, please try again
}
```

b. Endpoint 2: Add a New Book

- Endpoint URL: POST /api/books
- Description: Adds a new book to the database.
- Request Parameters: None
- Request Query: None
- Request body: Should be of the following format

```
{
  "title": "(str) - The title of the book, required",
  "author": "(str) - The author of the book, required",
  "edition": "(int) - The edition of the book, required",
  "publisher": "(str) - The publisher of the book, required",
  "language": "(str) - The language of the book, required",
  "publicationDate": "(str) - The publication date of the book
(YYYY-MM-DD), required",
  "numberOfCopies": "(int) - The number of copies available,
required"
}
```

- Response: Can be of types
    - 201 Created: Indicates successful addition of the new book.
    - 400 Bad Request: Occurs when the request is malformed or missing required parameters.
    - 500 Internal Server Error: Occurs when the server encounters an internal error or a database error.
- Notes
    - The response is an acknowledgement, indicating successful creation of a book.
    - If the request body has missing parameters or data in an incorrect format, a 400 response is returned.
    - If there is a database connection issue or unknown error, a 500 response is received with appropriate response.
- Examples:

1. Success Response (201 Created)

- Response Body:


```
{
    "status": "success",
    "info": "Book added successfully",
    "data": {
        "bookId": 32,
        "title": "title1",
        "author": "daniel1234",
        "edition": 1,
        "publisher": "Daniel123123",
        "language": "English",
        "publicationDate": "2023-03-02",
        "numberOfCopies": 10
    }
}
```


2. Error Response

- HTTP Status: 500 Internal Server Error
    - Response Body:

For database connection error


```
{
  "status": "error",
  "info": "Database connection error"
}
```


Unexpected errors


```
{
  "status": "error",
  "info": "An unknown error occurred, please try again later"
}
```

- HTTP Status: 400 Bad Request
    - Response Body:

      For Missing fields

```
{
    "status": "error",
    "info": "This field is required. for publisher"
}
```

      For duplicate candidate key

```
{
    "status": "error",
    "info": "The fields title, author, edition,
language, publisher must make a unique set. for
non_field_errors"
}
```

c. Endpoint 3: Update Book Details

- Description: Updates the details of a specific book in the database.
- Request Parameters:
    - Path Parameter: {id} - The unique identifier of the book to be updated.
- Request Query: None
- Request body: Should be of the following format

```
{
  "title": "(str) - The title of the book, required",
  "author": "(str) - The author of the book, required",
  "edition": "(int) - The edition of the book, required",
  "publisher": "(str) - The publisher of the book, required",
  "language": "(str) - The language of the book, required",
  "publicationDate": "(str) - The publication date of the book
(YYYY-MM-DD), required",
  "numberOfCopies": "(int) - The number of copies available,
required"
}
```

- Response: Can be of types:
    - 200 OK: Indicates successful update of the book details.
    - 400 Bad Request: Occurs when the request is malformed or missing required parameters.
    - 404 Not Found: Indicates that the specified book ID does not exist.
    - 500 Internal Server Error: Occurs when the server encounters an internal error or a database error.
- Examples:

1. Success Response (200 OK)

- Response Body:


```
{
    "status": "success",
    "info": "Book details updated successfully",
    "data": {
        "bookId": 9,
        "title": "title1",
        "author": "daniel1234",
        "edition": 1,
        "publisher": "Daniel",
        "language": "English",
        "publicationDate": "2023-03-02",
        "numberOfCopies": 10
    }
}
```


2. Error Response

- HTTP Status: 500 Internal Server Error
    - Response Body:

For database connection error


```
{
  "status": "error",
  "info": "Database connection error"
}
```


Unexpected errors


```
{
  "status": "error",
  "info": "An unknown error occurred, please try again later"
}
```

- HTTP Status: 400 Bad Request
    - Response Body:

      For Missing fields

```
{
    "status": "error",
    "info": "This field is required. for publisher"
}
```

      For duplicate candidate key

```
{
    "status": "error",
    "info": "The fields title, author, edition,
language, publisher must make a unique set. for
non_field_errors"
}
```

- HTTP Status: 404 Not found
    - Response Body:

```
{
    "status": "error",
    "info": "Book with bookId 2000 not found"
}
```