Project Documentation for Library Management System

Overview

The Library Management System streamlines the management of books, branches, user checkouts, and book transfers within a library network. It features a user-friendly interface and a REST API for various operations.

Objectives

- **Design and develop** a scalable Library Management System.
- **Implement core functionalities** for managing books, branches, user checkouts, and book transfers.
- **Ensure data integrity** with an optimized database schema.
- **Demonstrate best practices** in coding and project organization.

Technologies

- **Backend**: Node.js, Express.js
- **Database**: PostgreSQL
- **Frontend**: Next.js (Basic UI)
- **Tools**: Git (version control), Graphviz (ER diagrams)

Features

- **Books Management**:
  - **Create, Update, Delete**: Manage the library's book inventory by adding, updating, or removing books.
  - **Transfer**: Move books between different library branches.
  - **Filter & View**: Search for books by title, author, genre, branch location, or availability status.
- **Library Branches Management**:
  - **Create, Update, Delete**: Manage library branches by adding new branches, updating details, or removing them.
  - **Filter & View**: Find branches based on state or city.
- **User Management**:
  - **Create, Update, Delete**: Manage library users, including their association with specific branches.
  - **Filter & View**: Locate users by branch or other criteria.
- **Checkouts Management**:
  - **Create, Update, Delete**: Handle user book checkouts, assigning books to users manually.
  - **Filter & View**: Track checkouts by user ID and the date of the transaction.
- **Transfers Management**:
  - **Create, Update, Delete**: Manage the transfer of books between library branches.

Technical Side

The Library Management System uses a relational database schema built with Sequelize ORM to manage data efficiently. Below are detailed descriptions of each table in the schema:

- **Books Table**:
  - **Purpose**: Manages the library's book inventory.
  - **Columns**:

- `title` : The title of the book (required).
- `author` : The author(s) of the book (required).
- `genre` : The genre of the book (required).
- `quantity` : The number of copies available (required).
- `currentBranch` : The branch where the book is currently located (required).
- `branchId` : References the Branches table to track which branch holds the book (optional).
- `status` : Tracks the book's status (available, checked out, or reserved).
- `isbn` : The International Standard Book Number, a unique identifier for the book (required).

- **Branches Table**:
  - **Purpose**: Stores information about the library's branches.
  - **Columns**:
    - `name` : The branch's name (required).
    - `address` : The physical address of the branch (required).
    - `city` : The city where the branch is located (required).
    - `state` : The state where the branch is located (required).
    - `zipCode` : The postal code for the branch (required).
    - `phone` : The contact phone number for the branch (optional).

- **Checkouts Table**:
  - **Purpose**: Tracks the checkout process of books by users.
  - **Columns**:
    - `bookId` : References the Books table to indicate which book was checked out (required).
    - `userId` : References the Users table to track which user checked out the book (required).
    - `branchId` : References the Branches table to indicate from which branch the book was checked out (optional).
    - `checkoutDate` : The date the book was checked out (required).
    - `returnDate` : The expected or actual return date of the book (optional).
    - `status` : Indicates whether the book is checked out or returned.

- **Transfers Table**:
  - **Purpose**: Manages the transfer of books between branches.
  - **Columns**:
    - `bookId` : References the Books table to track which book is being transferred (required).
    - `fromBranchId` : References the Branches table to indicate the originating branch (required).
    - `toBranchId` : References the Branches table to indicate the destination branch (required).
    - `quantity` : The number of copies being transferred (required).
    - `transferDate` : The date the transfer took place (automatically set to the current date by default).

- **Users Table**:
  - **Purpose**: Stores user details and their association with branches.
  - **Columns**:
    - `name` : The user's full name (required).
    - `email` : The user's email address, which must be unique (required).
    - `phone` : The user's phone number (optional).
    - `role` : Defines the user's role within the library system (either librarian or member).
    - `branchId` : References the Branches table to associate the user with a specific branch (optional).

```
+----------------+        +----------------+        +----------------+        +----------------+
|   Branches     |        |     Books      |        |   Checkouts    |        |  Reservations  |
+----------------+        +----------------+        +----------------+        +----------------+
| id (PK)        |  1  --<| id (PK)        | >-- 1  | id (PK)        | >-- 1  | id (PK)        |
| name           |        | title          |        | bookId (FK)    |        | bookId (FK)    |
| address        |        | author         |        | userId (FK)    |        | userId (FK)    |
| city           |        | genre          |        | checkoutDate   |        | reservationDate|
| state          |        | quantity       |        | dueDate        |        | status         |
| zipCode        |        | currentBranch  |        | status         |        +----------------+
| phone          |        | branchId (FK)  |        +----------------+
+----------------+        | status         |
                          | isbn           |
                          +----------------+


+----------------+                                                            +----------------+
|     Users      |  1   --<  Checkouts/Reservations (See Above)   >-- 1       |   Transfers    |
+----------------+                                                            +----------------+
| id (PK)        |                                                            | id (PK)        |
| name           |                                                            | bookId (FK)    |
| email          |                                                            | fromBranchId (FK)|
| phone          |                                                            | toBranchId (FK)|
| createdAt      |                                                            | quantity       |
| updatedAt      |                                                            | transferDate   |
+----------------+                                                            +----------------+
```

Project Structure

- **Root Directory**:
  - `app.js` : Initializes the Express app.
  - `index.js` : Starts the server.
  - `package.json` : Project metadata and dependencies.
  - `seeders/` : Contains scripts for populating the database.
  - `src/` : Contains the main source code.
- `src/` Directory:
  - `config/` : Database configuration.
  - `controllers/` : Handles request logic.
  - `middleware/` : Custom middleware functions.
  - `models/` : Sequelize models for database tables.
  - `routes/` : API endpoint definitions.
  - `services/` : Business logic services.
  - `utils/` : Utility functions.
  - `validations/` : Input validation logic.

LIBRARY-MANAGEME...

- node_modules
- seeders
- src
  - bash
  - config
  - controllers
  - middleware
  - models
  - routes
  - services
  - utils
  - validations
  - .gitignore
  - app.js
  - index.js
  - package-lock.json
  - package.json
  - README.md