Design Document for Library Management System

1) Graphical User Interface (GUI)

- The Library Project's graphical user interface (GUI) was crafted using Django, a powerful web framework renowned for its efficiency, cleanliness, and practical design approach. Django significantly expedites the development process while ensuring a clean and robust interface. It simplifies communication between the database and models through its Object Relational Mapper (ORM), streamlining operations and reducing potential errors associated with database interactions.

2) Book search and availability

- In the database, books are uniquely identified by their **ISBN10** codes, and the columns ISBN10 and Title are mandatory fields, meaning they cannot be left empty.
- Authors are stored in a separate table called **'authors'**, where each author has a unique ID (**AuthorId**) and a name (**Name**), which is also unique in the table.
- To find a book using its **ISBN10**, the complete code must be entered; partial matches are not allowed.
- Searching for books by an author's name involves referencing a 'Book Authors' table, which connects books with their respective authors using foreign keys (Book **Author Id** and Book **ISBN10**). When a book is deleted from the parent table, the cascade on delete option handles the related entries in this table.
- When search results, details include the book's title, **ISBN10**, a comma-separated list of authors credited for the book, and its availability status. The status itself isn't stored as a separate column in the book model but is managed within the views.
- Additionally, the book search results offer a 'Loan Book' option. Clicking on book column tab directs users to a page where they can enter a card ID and proceed with loaning the desired books.

3) Book loans

- The Book Loans table in our Project comprises essential columns such as Loan Id (primary key), Card ID (foreign key to the Borrowers table with cascade on delete), ISBN (foreign key to the Books table with cascade on delete), Date Out (defaulted to the current date), Due Date (defaulted to 14 days after the current date), and Date In, which can be null or blank.

  For Checking Out Books:

  - To facilitate book loans, a user-friendly form is available. It includes fields for Card ID, ISBN10, Date Issued (Date Out in the model), and Due Date.

  - Users can search for a book from the home page and book tab Clicking redirects them to the Add Book Loans page, simplifying the process by allowing entry of the Card ID and a simple submit to initiate the loan.

Error handling:

- Invalid Card IDs trigger an error message: " Card-ID dose not exists."
- Users attempting to loan a fourth book when already having three active loans receive an error: "You already have 3 loans on your name."
- If a book is already on loan, an error is displayed: " This Book is not available."
- Loans are restricted for Card IDs with outstanding fines, prompting the error: " Card Id has some Fines..!."
- Automation: The Date Out defaults to the current date, while the Due Date automatically calculates 14 days ahead. The Loan ID is generated in the background.

For Checking In Books:

- The Book Check-Ins page allows users to perform book returns.
- Book loans can be searched using substrings of CardID, borrower's first or last name, or ISBN of a book.
- A user-friendly interface provides radio buttons for selecting the book to be checked in. Clicking
- the "Check In" button initiates the return process.

4) Borrower management

- **Card ID**: Automatically generated upon adding a new borrower but omitted from the borrower creation form for simplicity.
- **SSN**: A unique identifier for each borrower, imperative for creating a borrower record.
- **Borrower Name**: Split into Borrower First Name and Borrower Last Name, mandatory fields that cannot be left empty.
- **Address**: Comprising Address, City, and State, each mandatory for creating a new borrower record.
- **Phone Number:** An optional field that, if entered, should consist of exactly 10 digits without any characters.
- To create a new borrower, a form is available with fields for SSN, First Name, Last Name, Email, Street Number/Apartment, City, State (defaulted to TX), and Phone Number.

Error handling includes:

- Preventing the creation of a borrower with an existing SSN: "SSN already exists."
- Validating SSN length (should be at least 9 digits) and composition to avoid invalid SSNs.
- Ensuring mandatory fields (SSN, First Name, Last Name, and Address) are filled before creating a borrower; prompting users to complete these fields if left empty.
- Validating the phone number: it must be precisely 10 digits and free from characters. An error message will be displayed for an invalid phone number: " Please Enter Digits. "
- This approach emphasizes data integrity and user-friendly form interaction for creating borrower records, ensuring that mandatory fields are completed and validating inputs to maintain accurate borrower information in the library system.

5) Fine generation

- **Loan ID**: Acts as a foreign key to the Book Loans table with cascade action on delete.
- **Fine Amount**: Recorded as a decimal field.
- **Paid**: A Boolean field indicating the payment status.

The fines page offers several functionalities:

- **Filtering:** Fines can be filtered based on payment status - Paid, Unpaid, or displaying All fines.
- **Displaying Total Amount:** The table shows the total fine amount associated with a specific Card ID. This amount can be updated using the provided 'Update' button.
- **Marking as Paid**: Each row in the table has a 'Mark as Paid' button to update the payment status from unpaid to paid.

For enhanced functionality and data integrity:

- **Restrictions on Payment:** A validation process ensures that fines can only be paid for books that have been returned or checked in. Attempting to mark a fine as paid for a book that hasn't been returned triggers an error message: "Payment is not allowed for fines on books that are not yet returned."
- This approach ensures fines are effectively managed and displayed, allows easy filtering based on payment status, enables updating total amounts, and maintains data accuracy by preventing premature fine payments before book returns.