Documentation: Library Management System Code Explanation

1. `login.cpp`

This file serves as the entry point of the Library Management System application. It provides a menu for user login options and allows librarians to access the system's features.

Functions:

- `main()`: The main function where the program execution starts. It displays a menu for user login options and allows access to the librarian's interface upon successful login.

Variables:

- `choice` (int): Stores the user's choice for login options. It is used to capture the selected menu option.
- `flag` (bool): A boolean flag that controls the loop for multiple login attempts. It helps control whether the login attempts should continue or stop based on the user's input.

2. `LibrarianDef.h`

This header file defines data structures and functionalities related to book management in the library.

Data Structures:

- `Book`: A struct representing a book with various attributes such as serial number, book name, author, publish date, borrower ID, borrow date, due date, returned status, and status. This structure encapsulates information about each book.

Classes:

- `BookList`: A class representing a list of books with various book management methods. It acts as a container for managing a collection of `Book` instances.

Methods in `BookList`:

- `addBook(string serial, string bookname, string author, string publishdate)`: Adds a new book to the list with provided details. It creates a new `Book` instance and adds it to the list of books.
- `deleteBook(string serial)`: Deletes a book from the list based on the provided serial number. It searches for the book with the given serial number and removes it from the list.
- `writeToFile(string filename)`: Writes the book data to a specified file in CSV format. It iterates through the list of books and writes their details to the file.
- `modifyBook(string serial, string bookname, string author, string publishdate)`: Modifies the details of a specific book based on the provided serial number. It allows updating the book's name, author, and publish date.
- `borrowBook(string serial, string borrowerID, string borrowDate, string dueDate)`: Enables borrowing a book by updating its borrower information and due date. It sets the book's status to "Borrowed" and updates borrower-related details.
- `returnBook(string serial)`: Processes the return of a borrowed book. It updates the book's status to "Available," clears borrower information, and calculates fines if applicable.
- `listBorrowedBooksByBorrowerID(string borrowerID)`: Lists all books borrowed by a specific borrower based on their ID. It displays details of borrowed books for the provided borrower ID.

3. `Librarian.h`

This header file contains the librarian's login mechanism and functionalities to interact with the `BookList`.

Functions:

- `LibraryLogin()`: Implements the login process for librarians. It prompts for a user ID and password, validates them against a login data file, and grants access to librarian functions upon successful login.

- `addBooks(BookList &books)`: Allows librarians to add books to the library's collection. It interacts with the `BookList` instance to add new books.
- `deleteBooks(BookList &books)`: Enables librarians to delete books from the library's collection. It interacts with the `BookList` instance to remove books.
- `modifyBooks(BookList &books)`: Provides librarians with the ability to modify details of existing books. It interacts with the `BookList` instance to update book information.
- `borrowBooks(BookList &books)`: Allows librarians to process book borrowing transactions. It interacts with the `BookList` instance to handle borrowing operations.
- `returnBooks(BookList &books)`: Enables librarians to process book return transactions. It interacts with the `BookList` instance to handle book returns.
- `listBorrowedBooksByBorrowerIDs(BookList &books)`: Lists all books borrowed by a specific borrower based on their ID. It interacts with the `BookList` instance to display borrower-related details.

Variables:

- `userid` (string): Stores the user ID entered during the login process. It holds the librarian's entered user ID for validation.
- `password` (string): Stores the password entered during the login process. It holds the librarian's entered password for validation.
- `details` (string): Temporarily stores each line of user details from the login data file during reading. It assists in processing user details.
- `word` (string): Temporarily stores each word extracted from the `details` line. It is used to split the comma-separated values in each line.
- `l` (string): Temporarily stores each line read from the login data file. It holds a line of user details for processing.

- `users` (vector<string>): Stores user details read from the login data file. It holds the extracted user information for comparison during login.
- `fx` (int): A counter that tracks the number of incorrect login attempts. It is used to limit the number of login attempts before application exit.
- `books` (BookList): An instance of the `BookList` class that manages the library's collection of books. It is used to interact with the book management methods.

4. `date.h`

This header file provides a utility function for date validation.

Functions:

- `bool isValidDate(std::string date)`: Validates date strings in the "YYYY-MM-DD" format. It uses stringstream and `get_time` to parse and validate the date format.