

CS342 Project

Student's Names

Saad Al Nawmasi	ID:431108105
Mohammed Alhamat	ID: 431114833
Wael AlFaqr	ID: 431114842
Albarq AlFatehi	ID: 431114838
Suliman Al Duraibi	ID:431108067

Project Report: Online Library Management System

1. Application Overview

The "Online Library Management System" offers a role-based user experience with login and registration functionalities for both admins and general users. It connects to a Microsoft Access database, ensuring secure user management and data storage. Features include:

- Role-specific login and access.
- Registration of new users and admins with proper data validation.
- Smooth navigation between views for enhanced user interaction.
- **Book Management:** Admins can add, edit, remove, and track books.
- **User Management:** Admins can view, search, and manage user data.
- **Transaction Management:** Handles book borrowing and returns, including tracking overdue books.
- **Search Functionality:** Provides dynamic search options for books, users, and transactions.
- Providing users with a user-friendly interface to browse and borrow books.
- Displaying detailed user information, borrowed books, and transaction histories.
- Allowing users to edit their personal details and search for books dynamically.

2. GUI Elements

The **Log_Rig_View** and its nested **RegisterView** classes extend the application's GUI. Key elements are:

- **Login View:**
 - *Text Fields:*
 - Username for entering the user's name.
 - Password for entering a secure password.
 - *Buttons:*
 - **Login:** Verifies user credentials.
 - **Register:** Opens the registration view for new account creation.
 - *Purpose:* Enables users to log in based on their roles (Admin or User).
- **Register View:**
 - *Text Fields:*
 - Username, Password, Email, First Name, Last Name for capturing user details.
 - *Labels:* Associated with each text field for better usability.
 - *Buttons:*

- **Register:** Submits the form and saves the user's details into the database.
 - *Purpose:* Allows admins or users to create new accounts.
- **AdminWindow GUI Elements:**
 - **Buttons:**
 - **Add Book:** Adds a new book to the library's database.
 - **Remove Book:** Deletes a book by its title.
 - **Search User:** Filters users based on provided criteria.
 - **Return Selected Book:** Finalizes a book return transaction.
 - **Logout:** Logs out the current user and returns to the login screen.
 - **Text Fields:**
 - Input fields for book title, author, and available copies (Add Book).
 - Input field for book title (Remove Book).
 - Search fields for user information and transactions.
 - **Tables:**
 - **usersTable:** Displays user data dynamically.
 - **transactionsTable:** Shows transaction data related to book borrowing and returns.
 - **Panels and Frames:**
 - **JFrame:** Used for displaying individual windows (e.g., Add Book, Remove Book).
 - **JPanel:** Groups and organizes GUI components in each window.
- **LibraryWindow GUI Elements:**
- **Buttons:**
 - **My Information:** Displays the user's personal details.
 - **Edit Information:** Allows the user to update their personal information.
 - **Borrowed Books:** Shows a list of books borrowed by the user.
 - **Borrow:** Initiates the borrowing process for the selected book.
 - **Search:** Filters the displayed books based on the search query.
 - **Logout:** Logs the user out and returns them to the main application window.
- **Text Fields:**
 - **Search Field:** Accepts user input for book searches by title, author, or genre.
- **Tables:**
 - **Books Table:** Displays book details, including ID, title, author, genre, and availability.
 - **Borrowed Books Table:** Shows transaction details, including borrowed books and their due dates.
- **Toolbars:**
 - **Toolbar:** Provides quick access to user information, borrowed books, and borrowing functionality.
- **Backgrounded GUI Elements:**
 - A background added to the interface to .

3. Database Connection

The application uses SQL queries to interact with the Admins and Users tables in the Microsoft Access database. Connection details are passed to the login and registration views, ensuring:

- Secure communication via **Prepared Statements**, preventing SQL injection attacks.
- Role-based separation for admins and users through distinct database queries.
- Example queries:
 - INSERT for adding new books.
 - SELECT for retrieving user and transaction data.
 - DELETE for removing books and transactions.
 - UPDATE for modifying book availability.
- **Transaction Safety:**
 - Queries are encapsulated in try-catch blocks to ensure rollback in case of failures.
- **Fetching Book Details:**
 - **Query:** Retrieves books based on search criteria from the Books table.
- **Borrowing Books:**
 - Checks book availability.
 - Inserts a new transaction into the Transactions table with the borrowing details.
 - Updates the Books table to decrease the available copies count.
- **User Information Retrieval:**
 - Retrieves personal details from the Users table for display and editing.

4. Exception Handling

The code includes robust mechanisms to handle errors and ensure reliability:

- **Login Errors:**
 - Incorrect username or password prompts a failure message.
 - Failed attempts are caught and logged.
- **Registration Errors:**
 - Duplicate usernames or invalid data formats raise SQL exceptions, displayed to the user.
- **Database Issues:**
 - Missing or corrupted database file results in connection errors, handled gracefully.
- **Admin Window Errors**
 - **Possible Exceptions and Scenarios:**
 - **Database-Related:**
 - **SQLException:** Handled during database connectivity issues or invalid queries.

- Example: Displaying error messages for failed queries.
 - **Connection Failure:** Validates the database connection at startup and notifies the user.
- **Input Validation:**
 - Empty Fields: Alerts the user if required fields are left blank.
 - Invalid Data Types: Ensures numerical fields (like available copies) only accept valid numbers.
- **User Actions:**
 - No Selection: Prevents operations like returning books without selecting a transaction.
 - Missing Records: Handles cases where a book or user doesn't exist in the database.
- **LibraryWindow Errors:**
 - The application includes robust mechanisms to handle errors gracefully, providing meaningful feedback to users.
- *Handled Scenarios:*
- **Database Errors:**
 - **SQLException:** Handles connection failures, query errors, or result set processing issues.
 - **Error Feedback:** Alerts users to database-related issues with appropriate error messages.
- **Input Validation:**
 - Ensures valid due dates for borrowing operations.
 - Prevents empty or invalid input during searches or personal information updates.
- **User Actions:**
 - Prevents borrowing without selecting a book.
 - Handles empty search results gracefully.

5. Testing and Validation

- **Functional Testing:**
 - Verified that correct credentials allow access while incorrect credentials are rejected.
 - Ensured new registrations are reflected in the database.
- Verified individual features like book addition, removal, and user management.
- Ensured dynamic updates in tables for search functionalities.
- Verified book search, borrowing, and personal information features.
- Ensured proper database updates after transactions.

- **Validation Testing:**
 - Tested for empty fields in both login and registration forms.
 - Ensured secure password handling and validation of email formats.
- All core functionalities were verified to work without crashes or inconsistencies.
- Database operations reflected accurately in real-time.
- **Edge Case Testing:**
 - Verified behavior with non-existent usernames or mismatched passwords.
 - Checked for SQL injection attempts by entering malicious strings.
 - Checked edge cases such as empty inputs, maximum field lengths, and invalid inputs.
 - Tested edge cases, such as no available copies or invalid due dates.

6. Code Integration

- **Role-Based Login:**
 - The `handleLogin` method differentiates between admin and user roles.
 - Retrieves the user ID after a successful login for further use.
- Enables users to browse books dynamically.
- Facilitates borrowing transactions while updating the database.
 -
- **Registration:**
 - Saves user details into the appropriate table (Admins or Users).
 - Automatically includes the current date as the membership date.
 - Allows users to view and edit their details securely.
- **Navigation:**
 - Smooth transition between login, registration, and subsequent views (e.g., `AdminWindow` or `LibraryWindow`).
 - Admin users are directed to the `AdminWindow`.
 - General users are directed to the `LibraryWindow`.
 - Provides clear navigation for viewing borrowed books and other user-specific functionalities.
 - Ensures proper session termination upon logout.

