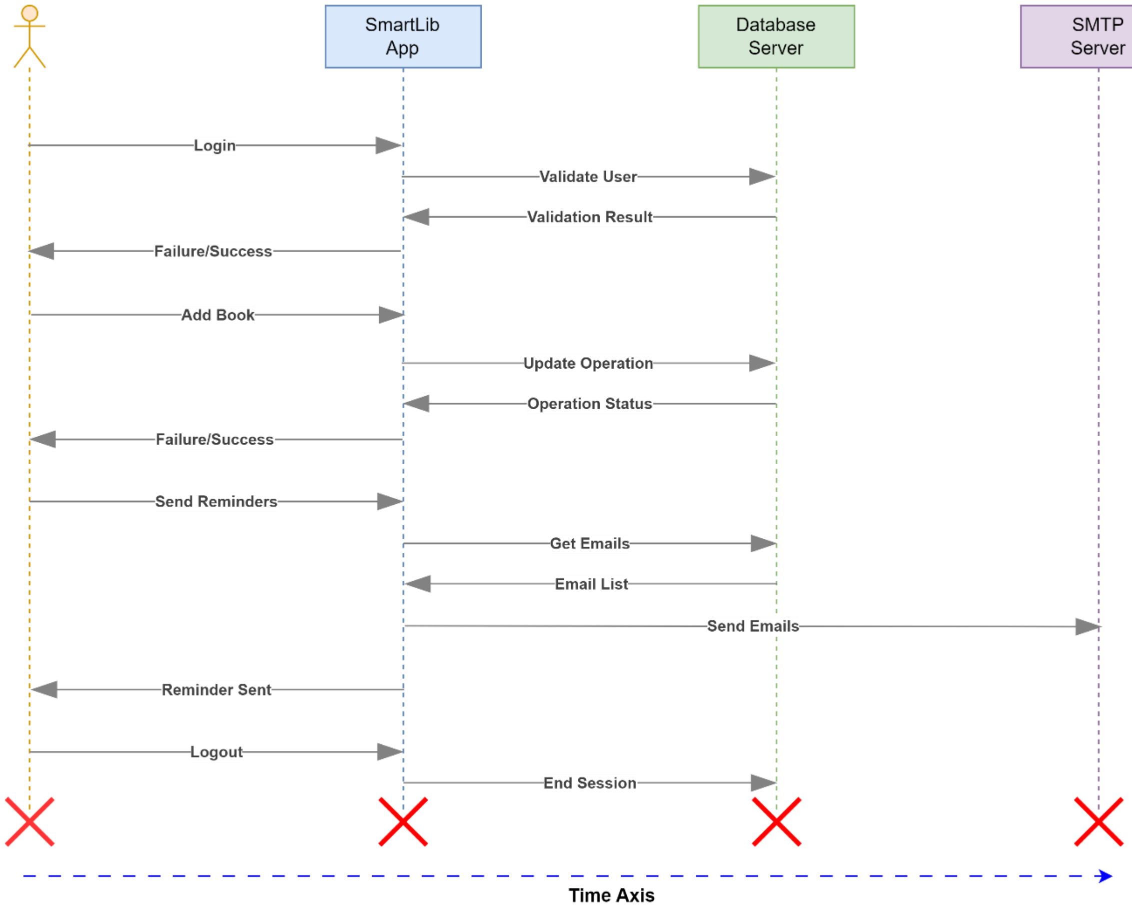
Timing Diagram



The timing diagram illustrates the sequence of interactions between different components of the **SmartLib system** over time. The components involved in this process are:

- **User**: The end-user interacting with the SmartLib system.
- **SmartLib**: The central application handling user requests and interfacing with both the database and the SMTP server.
- **Database**: The backend database responsible for storing and managing data related to books, borrowers, and transactions.
- **SMTP Server**: The email server used to send reminders and notifications to the users.

Each action in the system progresses from initiation by the **User** through handling by **SmartLib**, and potentially involves either the **Database** or the **SMTP Server** (or both) for additional operations. The timing diagram shows how these interactions are spaced out over time and how each component is engaged during these operations.

1. **Login Process:**
   - The **User** initiates a login request by providing credentials to **SmartLib**.
   - **SmartLib** forwards the login details to the **Database** for validation.
   - The **Database** validates the credentials and responds back to **SmartLib** with the result (successful or failed).

    - Based on the response from the **Database**, **SmartLib** sends a login success or failure message back to the **User**.

Time Considerations: This process typically happens within a few milliseconds, depending on the database response time.

2. **Book Operations (Add/Update/Delete Book):**
    - After logging in, the **User** can request to add, update, or delete book records.
    - **SmartLib** processes this request and communicates with the **Database** to update the relevant records.
    - The **Database** performs the requested operation and sends back an operation status (success/failure) to **SmartLib**.
    - **SmartLib** then relays the outcome of the operation to the **User**.

Time Considerations: The time for these operations may vary depending on the complexity of the data being modified, but the typical delay will be minimal, provided the database is responsive.

3. **Borrower Operations (Add/Update/Delete Borrowers):**
    - Similar to book operations, the **User** can manage borrower records (add/update/delete).
    - **SmartLib** forwards these requests to the **Database** to update the necessary records.
    - The **Database** completes the update and returns the result (operation status) to **SmartLib**.
    - The **User** receives feedback on whether the operation was successful or failed.

Time Considerations: These operations will be processed in a similar time frame as book operations, typically depending on the load and complexity of the database operations.

4. **Book Transaction Operations (Issue/Return/Renew Books):**
    - The **User** can also issue, return, or renew books through **SmartLib**.
    - For each of these requests, **SmartLib** updates the corresponding records in the **Database** (e.g., changing book status from available to issued).
    - Once the **Database** confirms the update, **SmartLib** sends the result of the transaction (success/failure) back to the **User**.

Time Considerations: These interactions follow the same pattern as other operations, with typical delays resulting from database query times.

5. **Sending Reminders (via SMTP Server):**
    - When the **User** requests reminder emails (e.g., for overdue books), **SmartLib** retrieves the relevant borrower information from the **Database**.
    - Once the **Database** returns the list of users with overdue books, **SmartLib** compiles an email list and sends it to the **SMTP Server** for processing.
    - The **SMTP Server** sends out the emails and returns a status confirmation (success/failure) to **SmartLib**.

Time Considerations: This process may take longer than other operations due to the involvement of external systems (SMTP Server) and the potential delay in sending multiple emails. The total duration can vary significantly depending on the volume of reminders.

6. **Logout Process:**

- The **User** initiates a logout request, and **SmartLib** terminates the session.
- Once the session ends, the system stops tracking the user's actions.

Time Considerations: This operation is generally instantaneous.