Library Management System - Visual Flow Diagrams

Document Information

Field	Value
Document Title	Library Management System - Visual Flow Diagrams
Version	1.0
Date	September 2024
Author	LMS Development Team
Status	Draft

System Architecture Overview

High-Level System Architecture

```
graph TB
    subgraph "CLIENT LAYER"
        A[Web Browser<br/>React.js]
        B[Mobile Web<br/>Responsive]
        C[Admin Panel<br/>React.js]
    end

    subgraph "APPLICATION LAYER"
        subgraph "Spring Boot Application"
            D[Controller Layer]
            E[Service Layer]
            F[Repository Layer]
        end
    end

    subgraph "DATA LAYER"
        G[MySQL Database]
    end

    A --> D
    B --> D
    C --> D
    D --> E
    E --> F
    F --> G
```

User Authentication Flow

Login Process Flow

```
sequenceDiagram
    participant U as User
    participant F as Frontend
    participant B as Backend
    participant D as Database

    U->>F: Enter credentials
    F->>F: Validate input
    F->>B: POST /api/auth/login
    B->>D: Check user credentials
    D-->>B: User data
    B->>B: Generate JWT token
    B-->>F: Token + User data
    F->>F: Store token in localStorage
    F->>F: Redirect to dashboard
    F-->>U: Show dashboard
```

Registration Process Flow

```
flowchart TD
    A[User visits Register page] --> B[Fill registration form]
    B --> C{Form validation}
    C -->|Invalid| D[Show validation errors]
    D --> B
    C -->|Valid| E[Send registration request]
    E --> F{Backend validation}
    F -->|Username/Email exists| G[Show error message]
    G --> B
    F -->|Valid| H[Create user account]
    H --> I[Send success message]
    I --> J[Redirect to login page]
```

Book Management Flow

Book Borrowing Process

```
sequenceDiagram
    participant M as Member
    participant F as Frontend
    participant B as Backend
    participant D as Database

    M->>F: Search for books
```

```
    F->>B: GET /api/books/search
    B->>D: Query books
    D-->>B: Book results
    B-->>F: Book list
    F-->>M: Display books

    M->>F: Click borrow button
    F->>B: POST /api/borrowings
    B->>D: Check book availability
    D-->>B: Availability status
    B->>D: Create borrowing record
    B->>D: Update book availability
    B-->>F: Success response
    F-->>M: Show confirmation
```

Book Return Process

```
flowchart TD
    A[Member views My Borrowings] --> B[Select book to return]
    B --> C[Click return button]
    C --> D{Check due date}
    D -->|Overdue| E[Calculate fine]
    D -->|On time| F[No fine]
    E --> G[Update borrowing status]
    F --> G
    G --> H[Update book availability]
    H --> I[Create history record]
    I --> J[Send confirmation]
```

Data Flow Architecture

Request-Response Flow

```
graph LR
    subgraph "Frontend"
        A[User Interface]
        B[State Management]
        C[API Client]
    end

    subgraph "Backend"
        D[Controller]
        E[Service]
        F[Repository]
    end

    subgraph "Database"
```

```
    G[MySQL Tables]
end

A --> B
B --> C
C --> D
D --> E
E --> F
F --> G
G --> F
F --> E
E --> D
D --> C
C --> B
B --> A
```

Component Interaction Flow

React Component Hierarchy

```
graph TD
    A[App.jsx] --> B[AuthProvider]
    B --> C[Router]
    C --> D[Navbar]
    C --> E[Routes]

    E --> F[Public Routes]
    E --> G[Protected Routes]

    F --> H[Home]
    F --> I[Login]
    F --> J[Register]

    G --> K[Admin Routes]
    G --> L[Member Routes]

    K --> M[AdminDashboard]
    K --> N[BookManagement]
    K --> O[ManageUsers]

    L --> P[MemberDashboard]
    L --> Q[BookSearch]
    L --> R[MyBorrowings]
```

API Endpoint Flow

REST API Structure

```
graph TB
    subgraph "Authentication API"
        A1[POST /api/auth/login]
        A2[POST /api/auth/register]
        A3[POST /api/auth/logout]
    end

    subgraph "User Management API"
        B1[GET /api/users]
        B2[GET /api/users/{id}]
        B3[PUT /api/users/{id}]
        B4[DELETE /api/users/{id}]
    end

    subgraph "Book Management API"
        C1[GET /api/books]
        C2[GET /api/books/{id}]
        C3[POST /api/books]
        C4[PUT /api/books/{id}]
        C5[DELETE /api/books/{id}]
        C6[GET /api/books/search]
    end

    subgraph "Borrowing API"
        D1[GET /api/borrowings]
        D2[POST /api/borrowings]
        D3[PUT /api/borrowings/{id}]
        D4[GET /api/borrowings/user/{userId}]
    end
```

Error Handling Flow

Global Error Handling

```
flowchart TD
    A[Error Occurs] --> B{Error Type}
    B -->|Validation Error| C[Show field-specific errors]
    B -->|Authentication Error| D[Redirect to login]
    B -->|Authorization Error| E[Show access denied]
    B -->|Server Error| F[Show generic error message]
    B -->|Network Error| G[Show connection error]

    C --> H[Update UI state]
```

```
    D --> H
    E --> H
    F --> H
    G --> H

    H --> I[Log error for debugging]
    I --> J[Continue application flow]
```

Security Flow

Authentication & Authorization

```
sequenceDiagram
    participant C as Client
    participant F as Frontend
    participant B as Backend
    participant D as Database

    C->>F: Request protected resource
    F->>F: Check localStorage for token
    F->>B: Request with JWT token
    B->>B: Validate JWT token
    B->>B: Check user permissions
    B->>D: Query user data
    D-->>B: User information
    B-->>F: Authorized response
    F-->>C: Display protected content
```

Database Relationship Flow

Entity Relationships

```
erDiagram
    USERS ||--o{ BORROWINGS : "has many"
    BOOKS ||--o{ BORROWINGS : "has many"
    BORROWINGS ||--o{ BORROWING_HISTORY : "has many"

    USERS {
        bigint id PK
        string username UK
        string email UK
        string password
        enum role
        timestamp created_at
        timestamp updated_at
    }
```

```
    BOOKS {
        bigint id PK
        string title
        string author
        string isbn UK
        text description
        int total_copies
        int available_copies
        timestamp created_at
        timestamp updated_at
    }

    BORROWINGS {
        bigint id PK
        bigint user_id FK
        bigint book_id FK
        date borrow_date
        date due_date
        date return_date
        enum status
        timestamp created_at
        timestamp updated_at
    }

    BORROWING_HISTORY {
        bigint id PK
        bigint borrowing_id FK
        bigint user_id FK
        bigint book_id FK
        enum action
        timestamp action_date
        text notes
        timestamp created_at
    }
```

Deployment Flow

CI/CD Pipeline

```
flowchart LR
    A[Code Commit] --> B[GitHub Actions]
    B --> C[Run Tests]
    C --> D{Tests Pass?}
    D -->|No| E[Notify Developer]
    D -->|Yes| F[Build Application]
    F --> G[Deploy to Staging]
```

```
    G --> H[Integration Tests]
    H --> I{Staging Tests Pass?}
    I -->|No| J[Rollback]
    I -->|Yes| K[Deploy to Production]
    K --> L[Health Check]
    L --> M[Monitor Application]
```

Performance Monitoring Flow

Application Monitoring

```
graph TB
    subgraph "Application"
        A[Request Processing]
        B[Database Queries]
        C[API Responses]
    end

    subgraph "Monitoring"
        D[Metrics Collection]
        E[Performance Analysis]
        F[Alert Generation]
    end

    subgraph "Actions"
        G[Auto-scaling]
        H[Cache Management]
        I[Database Optimization]
    end

    A --> D
    B --> D
    C --> D
    D --> E
    E --> F
    F --> G
    F --> H
    F --> I
```

User Journey Flow

Complete User Journey

```
journey
    title Library Management System User Journey
    section Registration
      Visit Home Page: 5: User
```

```
    Click Register: 4: User
    Fill Registration Form: 3: User
    Submit Registration: 4: User
    Receive Confirmation: 5: User
  section Login
    Visit Login Page: 4: User
    Enter Credentials: 3: User
    Submit Login: 4: User
    Access Dashboard: 5: User
  section Book Management
    Search Books: 4: User
    View Book Details: 5: User
    Borrow Book: 4: User
    View My Borrowings: 4: User
    Return Book: 4: User
  section Profile Management
    Update Profile: 3: User
    Change Password: 3: User
    View History: 4: User
```

These visual flow diagrams provide a comprehensive overview of the Library Management System's architecture, processes, and interactions. They serve as a visual reference for understanding system behavior and can be used for documentation, training, and system analysis.