RFID-Based Library Checkout

UID:23BCS14344

1. Requirement Analysis:

a). Functional Requirements:

- User Authentication:
  - Admin and Student login/signup.
- RFID Integration:
  - Books and student ID cards are tagged with RFID.
  - Arduino reads RFID tags and sends data to the application.
- Book Checkout/Return:
  - Students scan RFID cards (ID) and books to check out or return.
- Transaction History:
  - Logs of all issued/returned books with timestamps.
- Admin Panel:
  - Manage books, users, view transaction history, verify activity.
- Book Reservation System
  → Allow students to reserve books in advance.
- Overdue Notifications & Fine Management
  → Auto-calculate fines for overdue books and notify users via email/UI.
- Search & Filter Books
  → Enable search by title, author, genre, availability, etc.
- Multi-device Access
  → Admin and students can use the system on desktop or mobile browsers.

b). Non-Functional Requirements:

- Scalability
  → Ensure the system handles increasing numbers of books and users efficiently.
- Audit Logging
  → Track who performed what action and when (useful for admin-level traceability).

- Offline Mode (Optional)
  → Queue RFID scans and sync data when the backend reconnects.
- Role-based Access Control (RBAC)
  → Different permission levels for Admin, Librarian, and Student.
- Session Timeout & Auto-Logout
  → For enhanced security, especially on public terminals**.**

- Smooth Arduino-PC communication via Serial or USB.
- Real-time feedback and UI updates.
- Error handling for unregistered tags or invalid checkouts.
- Secure login system (hashed passwords, session management).

c). Hardware Requirements:

Consider:

- ESP32 instead of Arduino Uno
  → Allows Wi-Fi communication directly instead of Serial USB, removing the need for a direct PC connection.
- LCD Display (Optional)
  → Display status (e.g., "Book Issued Successfully") directly on the hardware.
- Buzzer or LED Indicator
  → Give immediate feedback for success/error scans
- Arduino Uno.
- RC522 RFID Reader.
- RFID Tags (for books and student cards).
- Jumper wires, Breadboard.
- USB cable to connect Arduino to PC.

d). Software Requirements:

- Arduino IDE (for RFID reader logic).
- Backend (C#/.NET Core or ASP.NET).
- Database (SQL Server / SQLite).
- Frontend (HTML, CSS, JS).
- Serial Communication (between Arduino and C# software).

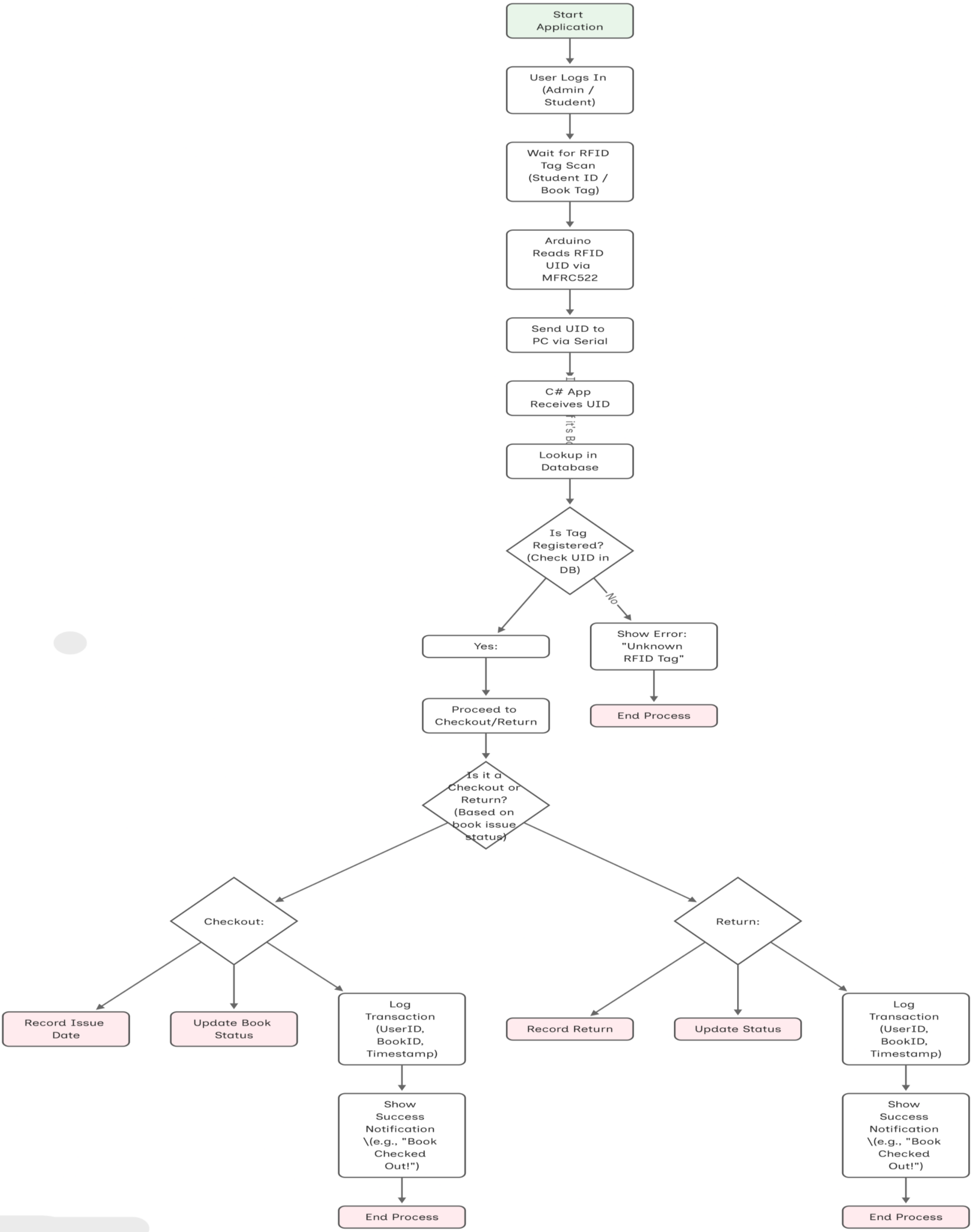
e). UI/UX Enhancements

- Responsive UI using Bootstrap or Tailwind CSS.
- Dark Mode Option for better accessibility.
- Live Dashboard for Admin:
  → Total books, currently issued books, real-time activity feed.

2. Objective

To design and implement a secure and efficient **RFID-based library system** using **Arduino hardware** for RFID reading and a **C# application** for login, user management, and real-time book checkout/check-in functionalities. This system aims to reduce manual book management and ensure accurate, tamper-proof logging of library transactions.

3. Flow Chart (Web Interface + Arduino + Backend)


Start Application
User Logs In (Admin / Student)
Wait for RFID Tag Scan (Student ID / Book Tag)
Arduino Reads RFID UID via MFRC522
Send UID to PC via Serial
I
C# App Receives UID
f it's Bo
Lookup in Database
Is Tag Registered? (Check UID in DB)
No
Yes:
Show Error: "Unknown RFID Tag"
Proceed to Checkout/Return
End Process
Is it a Checkout or Return? (Based on book issue status)
Checkout:
Return:
Record Issue Date
Update Book Status
Log Transaction (UserID, BookID, Timestamp)
Record Return
Update Status
Log Transaction (UserID, BookID, Timestamp)
Show Success Notification \(e.g., "Book Checked Out!")
Show Success Notification \(e.g., "Book Checked Out!")
End Process
End Process