1. Introduction

1.1. Project Overview

The **MetaMinds** Library Management System (LMS) aims to transform libraries into modern, user-focused environments that enhance accessibility and operational efficiency. This system will provide a comprehensive relational database for managing books, digital media, magazines, and user transactions. The LMS will support a freemium access model, allowing basic users to search and borrow materials while providing premium users with advanced analytics, custom notifications, and unlimited transactions. The system will streamline the check-out, return, and reservation processes, reducing wait times and improving library operations.

1.2. Scope

1.2.1. **Includes:**

- Maintain an extensive library catalog of books, magazines, and digital media.
- Support an efficient freemium model, offering different features based on user tiers.
- Enable advanced search and filtering capabilities, helping users locate resources quickly.
- Automate notifications and reporting, sending due date reminders and providing detailed analytics.
- Enhance staff efficiency by: streamlining checkouts, reservations, and overdue item management.

1.2.2. **Does not Include:**

- Direct e-book integrations (e.g., Kindle, Audible).
- RFID or barcode tracking for physical books.
- AI-based recommendations beyond basic borrowing history suggestions.

1.3. Glossary

1.3.1. **LMS** – Library Management System

1.3.2. **ISBN** – International Standard Book Number

1.3.3. **Freemium Model** – A system with both free and premium subscription tiers

1.3.4. **Supabase** – A backend-as-a-service providing Postgres, authentication, and APIs

1.3.5. **PostgreSQL (Postgres)** – The relational database system used for data storage

1.3.6. **RBAC** – Role-Based Access Control for managing permissions

1.3.7. **PCI DSS** – Payment Card Industry Data Security Standard

2. Stakeholders

2.1. Primary Stakeholders

2.1.1. **Library Patrons (Users):** Individuals who borrow and reserve books using the system.

2.1.2. **Library Staff**: Employees who manage book inventory, user registrations, and transactions.

2.1.3. **Library Administrators**: Responsible for overseeing system operations, memberships, and analytics.

2.2. Secondary Stakeholders

2.2.1. **System Developers & Database Administrators**: Maintain and optimize the database for performance and security.

2.2.2. **Third-Party Integrators**: External organizations that may integrate with the library system (e.g., educational platforms).

3. Requirements

3.1. Functional Requirements

3.1.1. **User Management**

- Users can register, log in, and manage accounts securely.
- Store user details, including membership type (free/premium), borrowing history, and contact information.
- Implement RBAC (Role-Based Access Control) to ensure appropriate permissions for users, staff, and administrators.

3.1.2. **Inventory Management**

- Store and manage books, magazines, and digital media with attributes such as:
  - **Books**: `Book_ID (PK), Title, Author, ISBN, Genre, Publication_Year, Availability_Status`
  - **Magazines**: `Magazine_ID (PK), Title, Issue_Number, Publication_Date, Availability_Status`
  - **Digital Media**: `Media_ID (PK), Title, Creator, Type, Genre, Availability_Status`
- The system updates the status when an item is checked out, reserved, or returned.

3.1.3. Advanced Searching and Filtering

- Users can search by title, author, ISBN, genre, publication year, and availability.
- Implement fuzzy search capabilities to account for minor spelling errors and partial matches.
- Provide sorting and filtering options (e.g., most borrowed, newest arrivals).

3.1.4. Transaction Handling

- Users can borrow and return books, with timestamps recorded.
- Borrowing limits enforced based on membership type:
  - **Free users**: Limited checkouts per month.
  - **Premium users**: Unlimited borrowing & reservations.
- Late returns are tracked, and late fees are automatically calculated.

3.1.5. Reservation and Hold System

- Users can reserve books currently checked out.
- Reservations have a validity period (e.g., 7 days) before expiring.
- The system notifies users when reserved books become available.

3.1.6. Freemium Model

Basic Tier (Free Users):

- Full access to the library catalog and search functionalities.
- Limited borrowing transactions per month.
- Basic notifications (due date reminders, availability updates).
- Access to community features (ratings, reviews, discussion forums).

Premium Tier (Subscription-Based Users):

- Unlimited borrowing and reservations.
- Advanced custom notifications (new arrivals, author-specific alerts).
- Detailed borrowing analytics and recommendations.
- Priority access to high-demand books.

3.1.7. Notification System

Basic Notifications:

- Due date reminders (via email/SMS).
- Overdue alerts with fine details.
- Reservation status updates.

Premium Custom Alerts:

- New book arrivals based on reading preferences.
- Upcoming library events and exclusive promotions.

3.1.8. Reporting and Analytics for Administrators

Generate real-time reports on:

- Borrowing trends by genre, author, and seasonality.
- Overdue books and fines collected.
- User engagement analytics for premium members.
- Library performance metrics, including most popular books.

3.2. Non-functional Requirements

3.2.1. Performance and Scalability

- Ensure search queries execute in <2 seconds, even with 100,000+ book records.
- Support at least 10,000 users, with an expansion plan for more.
- Handle at least 1 million transactions annually without performance degradation.

3.2.2. Security and Compliance

- Role-Based Access Control (RBAC) ensures proper permission levels for users and administrators.

- Sensitive user data (passwords, payment info) encrypted to meet PCI DSS & Kansas Data Privacy Laws compliance.
- Audit logging enabled to track system changes and prevent unauthorized access.

3.2.3. Reliability and Availability

- 99.9% uptime, leveraging Supabase-managed Postgres with automatic scaling.
- Automated daily backups to prevent data loss.
- Failover support for disaster recovery.

3.2.4. Usability and Accessibility

- Mobile-responsive design to allow users to browse and borrow books on mobile devices.
- Localization support for multiple languages.

3.2.5. Integration and API Support

- Public API access for external services (e.g., educational institutions).
- OAuth authentication for Google, Apple, and university login integration.

4. Development Resources Requirements

4.1. Software Requirements

Database Management System: PostgreSQL (Supabase)
Backend: Node.js / Python Flask
Frontend: Next.js / React.js
Hosting & Authentication: Supabase (for Postgres, authentication, and API services)
Deployment: Vercel or Firebase

4.2. Hardware Requirements

Cloud-based Supabase Postgres instance:

- **vCPU**: At least 2 cores (scalable)
- **Memory**: 4GB minimum, 8GB recommended
- **Storage**: 100GB+ with auto-scaling support

Local Development:

- 8GB RAM minimum, 16GB recommended
- SSD storage (256GB+) for development