## Architecture Requirements

### MySQL

I propose to use MySQL due to my familiarity with the platform and how to integrate it with my project. It will handle all information needed for the database aside from the images attached to books.

### Azure Service

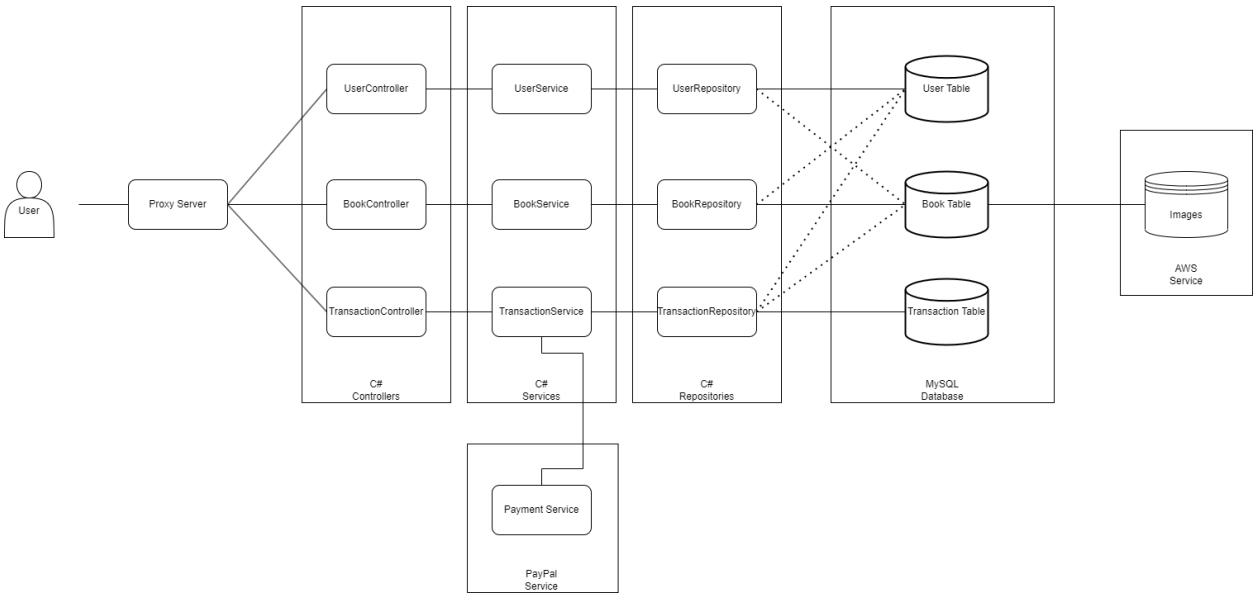
Azure supposedly has a user friendly integration with .NET products. This will be helpful to cut development time for Phase 3. Azure is also a Microsoft product, making it an interest for my other personal development in C#

### Web API (C#)

I am most versed in C# programming, making it the ideal for a public facing project. Should work hand and hand with Azure.

### PayPal

Most potential users that I know have a PayPal already set up, so no hassle for them. Protected platform without any shady practices.




Dates:

Phase 1 - 4/11/23
Phase 2 - 4/14/23
Phase 3 - 5/09/23

**Project Requirements**

• The system must store the profile of all users, including and not limited to: Username, first, last name, email address, and profile picture.
• The API must be able to store book information such as author, name, edition, description, ISBN, associating courses, book condition(new, used, etc.)
• The API must keep track of the seller( the student who posted the book for sale) and the buyer.
• The system must keep track of all interested patrons of a book and the winning patron.
• Payment transaction. You are required to use a third-party payment provider to handle all transactions. For this project, you will be using the sandbox mode.
• A filtering system should allow students to search books based on author, title, ISBN, and courses.
• The book can be removed from the database prior to being sold by the seller.
• The book can only be removed by the seller who owns the book.
• The interested patron can withdraw their interest in a particular book.
• The system must display a history of all books the seller had posted.
• The system must display a history of all books the buyer bought
• Sellers may make modifications to the book after it has been posted.
• The seller must upload images showing the condition of the book, and those images must be stored on a cloud provider data storage such as AWS s3 bucket, Azure blob, etc

## External Resources

OpenLibrary
https://openlibrary.org/developers/api

*Partner API* - Search ISBN: This API allows partners to search for books based on ISBN numbers. The API returns a JSON object with details about the book, including author, title, publisher, and more.

Microsoft Azure Blob Storage
https://azure.microsoft.com/en-us/products/storage/blobs

*Description* - "Microsoft Azure Blob Storage is a highly scalable and cost-effective cloud storage service provided by Microsoft Azure. It offers different storage tiers optimized for various access patterns, including hot, cool, and archive storage. Azure Blob Storage provides advanced security features such as role-based access control, encryption, and firewall protection. It can be integrated with other Azure services and third-party applications, making it a popular choice for businesses of all sizes"

PayPal
https://developer.paypal.com/api/rest/

*Standard Checkout*

Description - "Set up standard payments to present payment buttons to your payers so they can pay with PayPal, debit and credit cards, Pay Later options, Venmo, and alternative payment method"

**Data Requirements**

*User Data* - username, first and last name, email address, and profile picture
- Seller
- Buyer
- Patron

*Book Data* - author, title, edition, description, ISBN, associated courses, book condition, and seller information, Patron

*Transaction Data*

*Documentation*

User Data: The API should store user data, including their username, first and last name, email address, and profile picture. It should also store authentication and authorization information, such as the user's password and access tokens.

Book Data: The API should store book data, including the book's author, title, edition, description, ISBN, associated courses, book condition, and seller information. It should also store images of the book's condition.

Patron Data: The API should store patron data, including interested patrons for each book and the winning patron.

Transaction Data: The API should store transaction data, including payment information, book sale history, and user purchase history.

Documentation Data: The API should store data related to documentation, such as the API documentation, endpoints, and parameter

**Class Diagram (ER)**