

## **Project Requirements**

### **Swap with architecture**

"The purpose of the final project is to showcase your knowledge and skills gained in the course throughout the semester, which is cumulative work from the first day of class to the last day of class. This project is a collaborative group project, simulating real-world corporate life where each group member does specific tasks delegated and works together collaboratively to reach the common end goal: a functional and robust final product.

You will work on this project in parallel with NMAD 262, developing a REST API server for this course. Your API server will handle all requests from the front-end, including but not limited to processing data, storing data, fetching data, inquiring additional databases from an external API resource to supplement, storing data with a third-party cloud provider. Your product must incorporate the API best practices such as exception handlings, data validations, and among other things."

#### Must Have:

- Documentation (Spec. sheet, Class Diagram, Resources, etc)
- MySQL Data Server
- Azure Data Service - Images
- RESTful API (C#/.NET)
- Presentation

#### Dates:

Phase 1 - 4/11/23

Phase 2 - 4/14/23

Phase 3 - 5/09/23

## **Architecture Requirements**

### **Diagram goes here (Show services to use)**

- 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>

Book API - Search Book

Search API - Search Params

Partner API - Search ISBN

*Book API - Search Book:* This API allows users to search for books available in the Open Library database. Users can search by author, title, subject, and more. The API returns a JSON object with details about the books matching the search criteria, including book ID, author, title, publisher, publication date, and cover image.

*Search API - Search Params:* This API allows users to search for books based on specific search parameters, such as author, title, subject, and more. The API supports a wide range of search parameters and returns a JSON object with details about the books matching the search criteria.

*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"

## 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)

