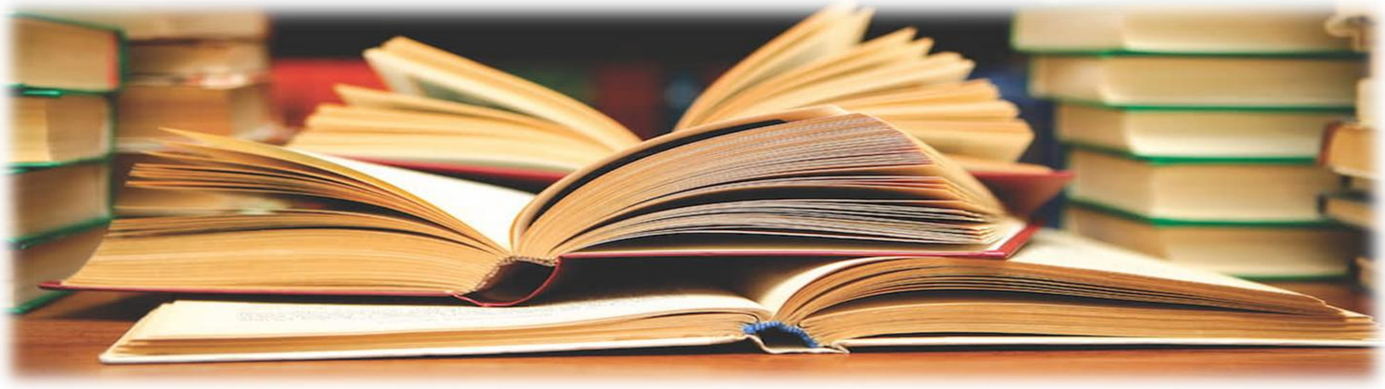
The Islamic University in Gaza – IUG

Faculty of Engineering

Computer Engineering Department

BOOK POINTS



Preparing by:

Nourhan Ihab Abd AL Aziz	**220210488**	**Abeer Ayman Alsibakhi**	**220210490**
Noor Jamal El Kord	**220212032**	**Reem Ehab Afana**	**220211284**
Aya Rami Frena	**220210510**	**Abrar Dyab Hajjaj**	**220213091**
Deema Wail Badah	**220212709**	**Shahed Harby Alhaddad**	**220210923**

Supervised by:

Dr. Ahmed Mahdi

Eng. Dina Harara

August/2025

1. Requirements

a) Functional Requirements

- The system shall allow users to register and log in using email and password to create and access personal profiles.

- The system shall enable users to list physical books or academic software (via USB metadata) for exchange, including details like title, author, course code, and optional photos.

- The system shall implement a points system where users earn points for listing items and deduct points for requesting exchanges.

- The system shall provide search functionality for books and software by title, or type (book/software).

- When requesting an item, the system shall show the owner’s contact details (name + contact method: email/phone).

- The system shall allow users to confirm completed exchanges, updating points and item status.

- The system shall display user profiles showing listed items, points balance, and exchange history.

b) Non-functional Requirements

- The system shall support Arabic language for user interface and content.

- The system shall ensure data security, including encrypted passwords and secure messaging.

2. Features

- Unified Role (Recipient & Provider).
- Choose your college and nearest meeting point.
- View available books and flash drives in your area.
- Search for a book or flash drive by title.
- Filter by resource type (book or flash drive).
- Add a book or flash drive to share.
- Reserve an item with a single tap (+).
- Cancel a reservation when needed (–).
- See the provider's name and phone number after reservation.
- View My Items (what you shared) and My Reservations.

3. User Stories:

User Story 1: Registration and Login

As a student, I want to register and log in so that I can access my personal profile and use the app.

User Story 2: Unified Role

As a student, I want to both share and request resources so that I can contribute and benefit from the system.

User Story 3: Resource Browsing

As a student, I want to search, filter, and view available resources so that I can quickly find books or flash drives I need.

User Story 4: Resource Sharing

As a student, I want to add a resource (book or flash drive) so that others can request it.

User Story 5: Reservation Management

As a student, I want to reserve or cancel a resource so that I can manage my borrowing easily.

User Story 6: Points System

As a student, I want to track points earned by sharing so that I can use them to borrow another item.

4. Scenarios :

1. Adding a Book :

User: Ahmed, a registered student.

Steps:

1. Ahmed logs into his account.

2. He selects his geographical location (e.g., Gaza City, Khan Younis, etc.).

3.He chooses his college from the list.

4. He clicks on "Add Book."

5. He enters the book title, author, and course code, then uploads a photo.

6. He submits the form.

7. The system adds the book to his profile and rewards him with points.

2. Borrowing a Book :

User: Sarah, a registered student.

Steps:

1. Sarah logs into her account.

2.She selects her geographical location.

3. She chooses her college from the list.

4. She searches for the required book or flash drive by title or course code.

5.The system filters results based on location college and shows availability.

6. She clicks "Request Exchange" and confirms.

7. The system deducts points, notifies the book owner, and updates her request list.

5. Specifications (Tasks):

1. As a student, I want to register and log in so that I can access my personal profile and use the app.

Tasks:

- Design a registration form (fields: name, email, password).
- Implement input validation (unique email, strong password).
- Create login functionality with email & password authentication.
- Store user details securely in the database.
- Redirect users to their profile after successful login.

2. As a student, I want to both share and request resources so that I can contribute and benefit from the system.

Tasks:

- Implement role flexibility (every user can be both provider and recipient).
- Add “Share” button to upload resources.
- Add “Request” button to reserve resources.
- Ensure shared items appear in the resource list for others to view

3. As a student, I want to search, filter, and view available resources so that I can quickly find books or flash drives I need.

Tasks:

- Create a resource list page.
- Implement search bar (search by title/keyword).
- Add filter option (Book / Flash Drive).
- Display resource details (title, type, availability).
- Show “No results found” message if nothing matches.

4. As a student, I want to add a resource (book or flash drive) so that others can request it.

Tasks:

- Design "Add Resource" form (fields: title, type, description, optional image).
- Save new resources to the database.
- Display added resources in the browsing list.
- Link shared resources to the provider's profile.

5. As a student, I want to reserve or cancel a resource so that I can manage my borrowing easily.

Tasks:

- Implement "Reserve" button on resource details.
- Mark reserved items as "Unavailable" for others.
- Allow users to cancel reservations.
- Update availability status in real-time.

6. As a student, I want to track points earned by sharing so that I can use them to borrow other items.

Tasks:

- Define points rules (e.g., earn points when sharing).
- Update points balance after each transaction.
- Display points in user profile.
- Deduct points when reserving them.

Acceptance Criteria:

1. College & Location Selection .

- User must be able to select their college from a predefined dropdown list.
- User must select a nearby meeting point from a predefined list
(e.g. College Library, University Gate).
- These selections must filter the resources shown throughout the app.

2. Receive Resource Flow

- User can choose between “Book” or “Flash Drive” after selecting
“Receive “.
- A search bar must be visible and functional to filter available resources.
- Only resources matching the user’s selected college and location should be shown.
- When the user taps on the "+" button:
    - The resource must be booked instantly.
    - A confirmation message must appear with.
    - Success icon and message.
    - Meeting location.
    - Time (e.g. 12 PM).
    - Donor's name and phone number.
    - Booked resources should appear in the user's “My Resources” section.

3. Add Resource Flow

- User can choose to add either a Book or a Flash Drive.
- User must enter the name of the resource (description is optional).
- After confirming, the resource must become visible to other users in the same college/location.
- When another user books the resource:

    The donor must receive an in-app notification.

    Notification must include the recipient's name and phone number.

4. User Profile Page

User must be able to view their:

- Earned points.
- Resources they added.
- Resources they book.
- Points must be updated correctly after each successful donation.

5. Search & Filtering

- Search bar must filter resources by keyword (name of book or flash).

- Only resources relevant to selected college and location must be shown.

6. Booking System

- Booking must be instant with no manual approval from the donor.

- Once booked, the same resource must not be visible to other users.

7. Notifications

- Donors must receive real-time in-app notifications when someone books their resource.

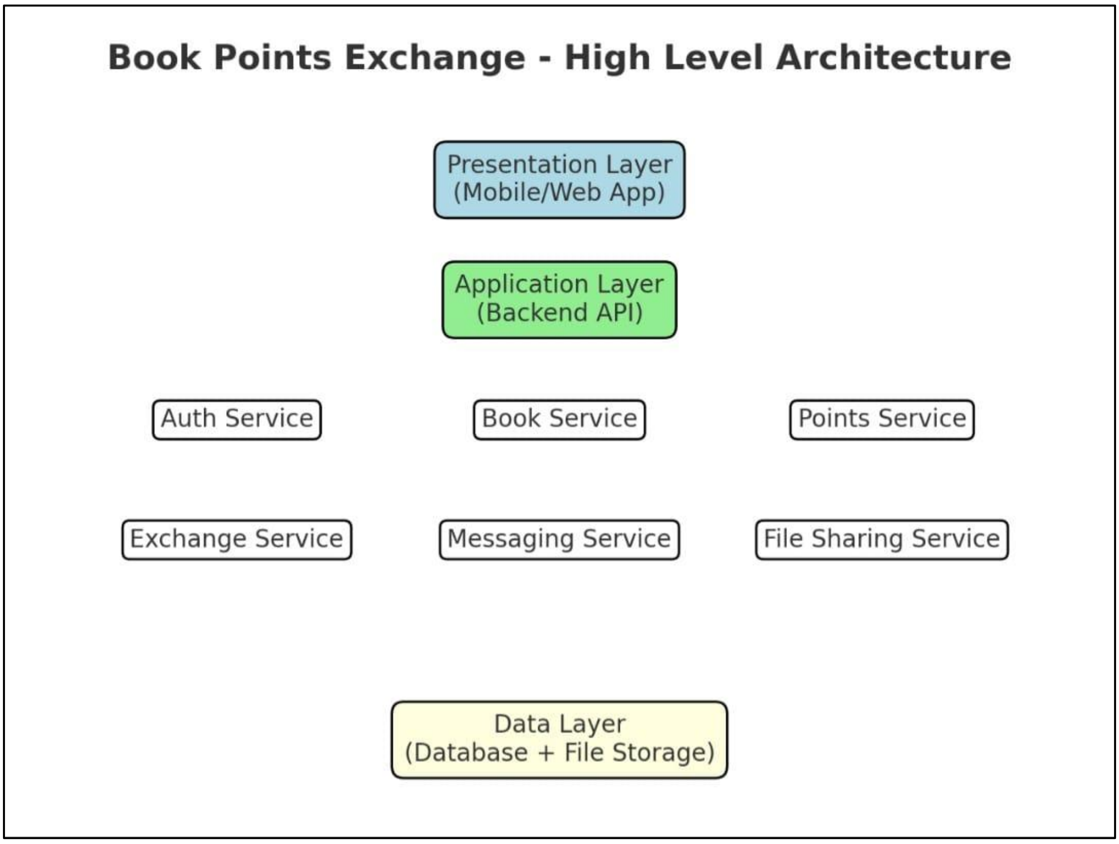
- Notifications must include full name and phone number of the recipient.

8. Points System

- Each successful donation should increase the user's points.

- Points should be clearly displayed on the profile page.

7. Software Architecture (Diagrams)


Book Points Exchange - High Level Architecture
Presentation Layer (Mobile/Web App)
Application Layer (Backend API)
Auth Service
Book Service
Points Service
Exchange Service
Messaging Service
File Sharing Service
Data Layer (Database + File Storage)

1- High-Level Architecture (Layers)

Layers :

Presentation Layer (Mobile App / Web App)

User Interface :

(browse books, add book, view points, messaging, file-sharing option)

Application Layer (Backend Services / APIs)

Authentication Service.

Book & Program Management Service.

Points & Transactions Service.

Notification Service.

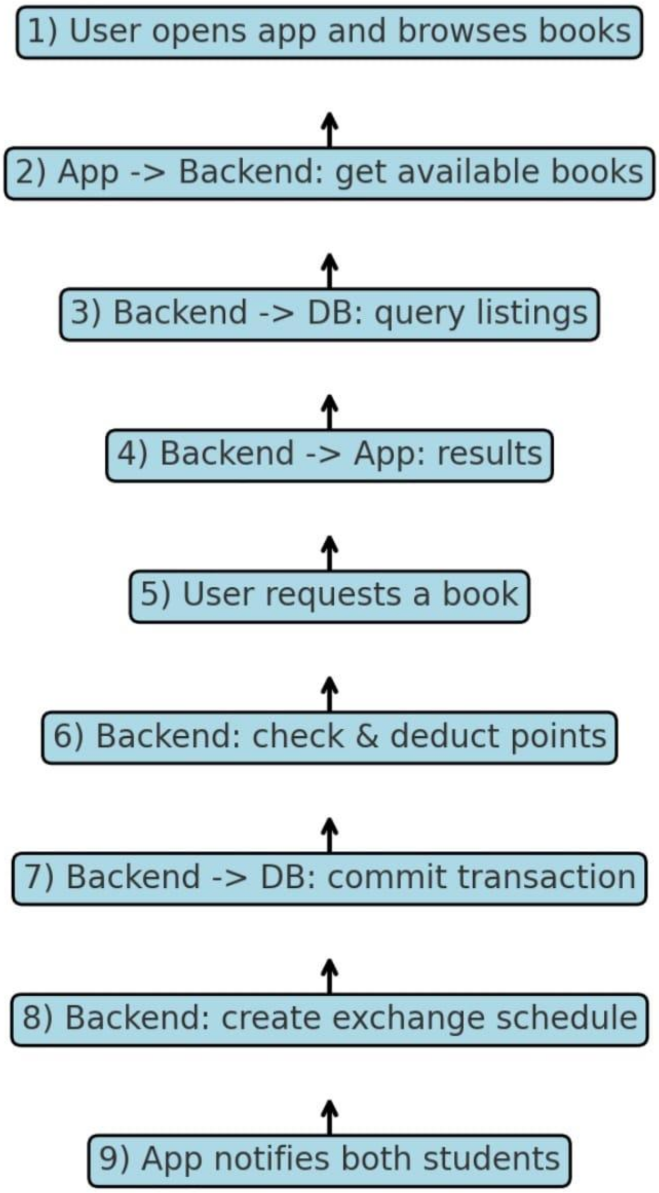
Exchange Management (location & scheduling).

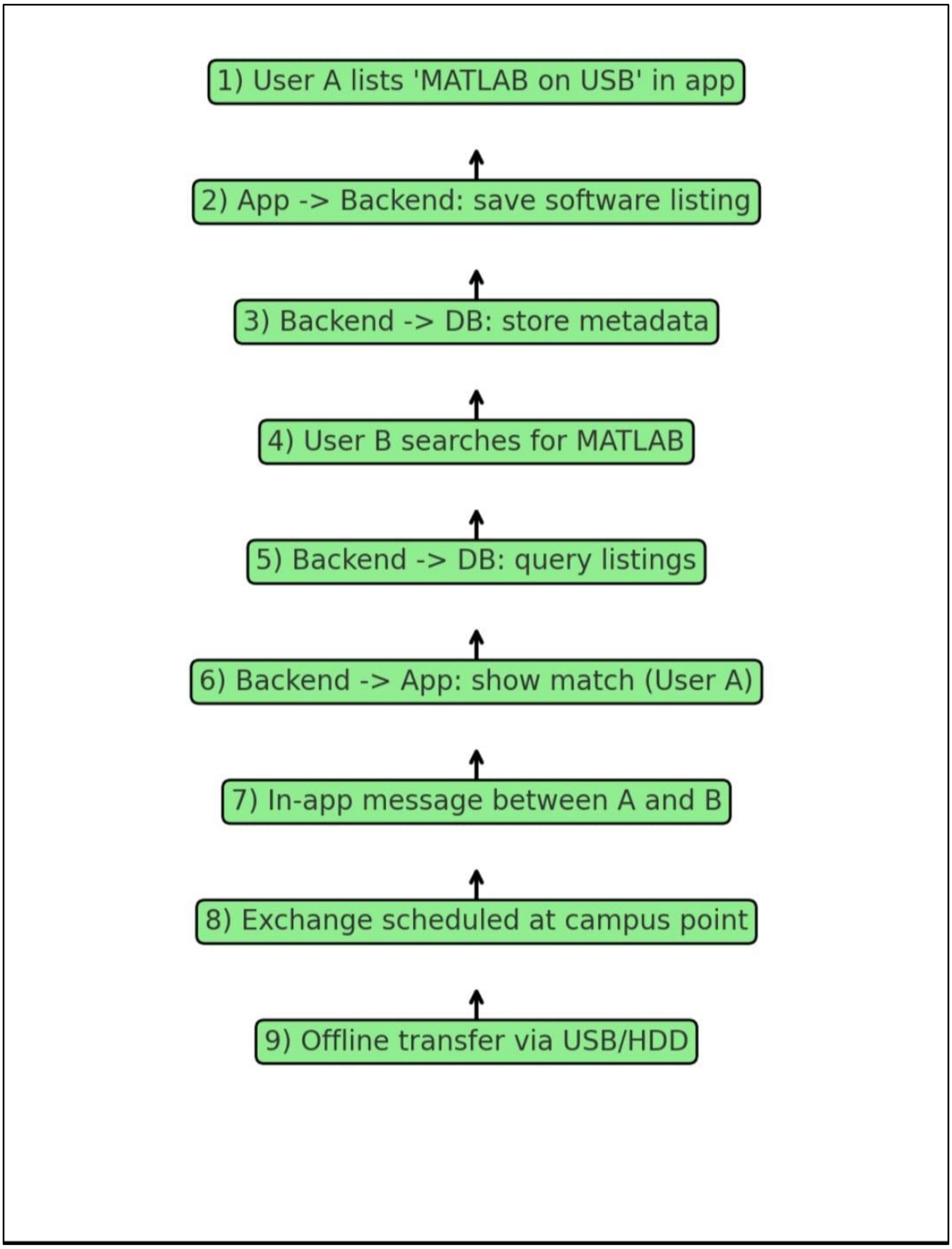
Data Layer (Database + File Storage)

User Data.

Books & Software Metadata.

Transactions & Points History.


1) User opens app and browses books
2) App -> Backend: get available books
3) Backend -> DB: query listings
4) Backend -> App: results
5) User requests a book
6) Backend: check & deduct points
7) Backend -> DB: commit transaction
8) Backend: create exchange schedule
9) App notifies both students


1) User A lists 'MATLAB on USB' in app
2) App -> Backend: save software listing
3) Backend -> DB: store metadata
4) User B searches for MATLAB
5) Backend -> DB: query listings
6) Backend -> App: show match (User A)
7) In-app message between A and B
8) Exchange scheduled at campus point
9) Offline transfer via USB/HDD