Workshop N° 1 - Project Definition and Database Modeling

Universidad Distrital Francisco José de Caldas
School of Engineering
Computer Engineering Program

David Stiven Muñoz Amaya - 20202020071

Janeth Oliveros Ramírez - 20182020100
Daniel Felipe Paez Mosquera - 20202020070

September 2025

1 Introduction

This project proposes the development of an Academic Digital Library Platform designed to complement, rather than replace, the university's physical library. The platform offers exclusive access for students and faculty, enabling DRM-free consultation and downloads of institutional content such as theses, research papers, and authorized digital books, while licensed materials from external providers (e.g., IEEE, Elsevier, Springer) are indexed with redirection.

Users have distinct roles: students search, consult, download, and review materials; professors access resources and suggest acquisitions; administrators manage catalogs, users, and reports; and optional digital librarians validate content and oversee licenses. Core features include advanced search, personalized recommendations, user reviews, and BI reports. The system emphasizes scalability, performance, and security, ensuring reliable 24/7 academic access and fostering research and learning excellence.

2 Canvas Business Model

Key Partners

- Academic publishers like IEEE, Elsevier, Springer.
- Cloud service providers
- University IT department
- Librarians and faculty validators

Key Activities

- Digitization and cataloging of institutional books.
- Integration with external providers.
- User and role management.
- Data processing for BI and recommendations.

Key Resources

- Scalable servers and databases.
- Publisher licenses.
- Security infrastructure.
- Development and support team.
- Analytics and BI system.

Value Propositions

- Digital platform that complements the physical library.
- Free consultation and downloads
- Advanced search functionality.
- Personalized recommendations.
- Academic citation export.

Customer Relationships

- Personalization based on history and degree program.
- Self-service like search, download or reviews.
- Technical and digital librarian support.

5. Channels

- Institutional web portal.
- Authenticated access with university email.
- Email notifications.

Customer Segments

- University students.
- Faculty members.
- Administrators.
- Professors.
- Digital librarians.
- Researcher.

Cost Structure

- Cloud or on-premise infrastructure.
- Publisher licenses.
- Salaries for technical staff, customer service and librarians.

Revenue Streams

- Fast and secure access to bibliography.
- Cost savings in duplicated physical materials.
- Increased university prestige and visibility.
- Support for research and teaching.

3 Requirements Documentation

3.1 Functional Requirements

Scope reminder. Digital University Library focused on students and professors. Resources are books and articles only. Downloads are DRM-free. Hybrid access model: (i) local university content is stored and downloadable; (ii) licensed/commercial content is indexed and redirects to the provider.

1. **User Registration and Authentication.** Users (Student, Professor, Admin, Librarian) must authenticate with institutional email (SSO or local login).
2. **Role-Based Access Control (RBAC).** System enforces role permissions for search, upload requests, approvals, and administration.
3. **Resource Ingestion (Local).** Admins and Librarians can create a resource with mandatory metadata and upload files (PDF/ePub).
4. **Upload Request (Professor Draft).** Professors can submit a draft resource for review (metadata + file or external link).
5. **Curation/Approval Workflow.** Admin/Librarian can approve, reject, or request changes with reason.

6. **External (Licensed) Resource Registration.** System supports records that store metadata and an `external_url` (no local file).
7. **Metadata Management.** CRUD for metadata (authors, affiliations, keywords, ISBN/DOI, abstract, language, license).
8. **Advanced Search & Browse.** Search by title, author, keywords; filters by year, type (book/article), language.
9. **Resource Details & Download.** Resource page shows full metadata, availability (local/external), and download or access button.
10. **Recommendations (Near Real-Time).** Show "You may also like" based on recent views/downloads and shared keywords.
11. **Reviews & Ratings.** Students/Professors can rate (1–5) and review resources; one review per user-resource.
12. **Citation Export.** Export APA/MLA and BibTeX for books and articles.
13. **Notifications.** Email/onsite notifications for workflow events (draft submitted, approved/rejected).
14. **BI Reports (Batch).** Nightly jobs compute: most consulted by program/semester/year, top authors, trending topics.
15. **Concurrency Control for Licensed Limits (Optional).** If a licensed resource has max concurrent accesses, system enforces limit.

3.2 Non-Functional Requirements

1. **Performance.** Searches should return results in less than 1 second for indexed queries. Opening a resource page should be fast (under 2 seconds).
2. **Scalability.** The system must support growth in the number of users and resources each semester (e.g., from hundreds to a few thousands).
3. **Availability.** The system should be available during the semester with minimal downtime. 24/7 availability is expected, but maintenance windows are acceptable.
4. **Security.** Access must be restricted by user role (student, professor, admin). Only authorized users (with institutional email) can log in and download resources.
5. **Data Management.** Store resources in a relational database. Optionally complement with a NoSQL or search engine for fast queries.
6. **Privacy.** Only minimal personal data should be stored (name, institutional email, role). Protect intellectual property of resources (local vs. external access).

4 User Stories

US1 - Search for resources

Priority: High **Estimate:** 5 points

User Story:
As a student, I want to search for books and articles by title, author, or keywords so that I can quickly find the material I need.

Acceptance Criteria:

- Given I am on the search page, When I enter a title, author, or keyword and click search, Then the results show matching resources with metadata and availability within 1 second.

US2 - Download resources

Priority: High **Estimate:** 3 points

User Story:
As a student, I want to download books and articles in PDF/ePub format so that I can study them offline.

Acceptance Criteria:

- Given I have selected a resource, When I click the download button, Then the file downloads in PDF/ePub and the action is logged with user, resource, and timestamp.

US3 - Leave reviews and ratings

Priority: Medium **Estimate:** 3 points

User Story:
As a student, I want to leave reviews and ratings for the resources I use so that I can share my opinion with other students.

Acceptance Criteria:

- Given I have accessed a resource, When I submit a review (max 500 characters) and a rating (1-5 stars), Then the review is displayed with reviewer name and date.

US4 - Professors access resources

Priority: High **Estimate:** 5 points

User Story:
As a professor, I want to access the digital books and articles available in the library so that I can use them in my classes and research.

Acceptance Criteria:

- Given I am logged in as a professor, When I search for and download resources, Then the system grants access only to authenticated users.

US5 - Suggest acquisition

Priority: Medium **Estimate:** 5 points

User Story:
As a professor, I want to suggest the acquisition of a book or article not available in the catalog so that the library can evaluate including it.

Acceptance Criteria:

- Given I searched and found no matching resource, When I click “Suggest acquisition” and fill the form (title, author, ISBN/DOI, justification), Then the suggestion is saved with status = Pending.

US6 - Validate and approve uploads

Priority: High **Estimate:** 8 points

User Story:
As a librarian, I want to validate and approve uploaded digital resources so that only authorized materials are published.

Acceptance Criteria:

- Given a resource is uploaded, When I review it as librarian, Then I can approve or reject it with a reason, and approved resources become visible.

US7 - Manage external resource licenses

Priority: Medium **Estimate:** 5 points

User Story:
As a librarian, I want to manage licenses of external digital resources so that users always have access to subscribed journals.

Acceptance Criteria:

- Given a resource has a license, When the expiration date is within 30 days, Then the system notifies the librarian and flags expired resources.

US8 - Manage user accounts and roles

Priority: High **Estimate:** 8 points

User Story:
As an administrator, I want to manage user accounts and assign roles (student, professor, librarian) so that access is secure and controlled.

Acceptance Criteria:

- Given I am an admin, When I create, update, or deactivate a user, Then only predefined roles can be assigned and all actions are logged.

US9 - Manage digital catalog

Priority: High **Estimate:** 5 points

User Story:
As an administrator, I want to manage the digital catalog so that the content available to users is always updated.

Acceptance Criteria:

- Given I am an admin, When I add, update, or remove a resource, Then the changes are reflected immediately in the catalog.

5 Initial Database Architecture

- **High-Level Architecture Overview:**
  The database system for the Digital University Library is designed under a **4-layer architecture** model, supporting both transactional and analytical workloads, with scalability and high availability as priorities to handle big data and distributed system requirements.

  1. **Presentation Layer:**
     - User interface (web) interact with the backend services via REST APIs or GraphQL.
  2. **Application Layer:**
     - Manages business logic, request orchestration, authentication, and communication with the database services.
  3. **Domain Layer:**
     - Encapsulates the core domain entities (books, users, reviews, loans) and enforces business rules before data is persisted.
  4. **Infrastructure Layer (Data Storage & External Services):**
     - **Relational Database:** *PostgreSQL* is used as the main transactional database to store structured data such as users, resources, reviews, and loans.
     - **Object Storage:** For storing large digital files (e.g., PDF, ePub), an open-source object storage solution like *MinIO* is proposed.
     - **Search Engine:** *OpenSearch* (community-driven fork of Elasticsearch) is integrated to provide fast full-text search and filtering capabilities over a large catalog.

- **Big Data and Distributed System Considerations:**
  - The infrastructure supports replication, partitioning, and backup to ensure data availability and fault tolerance.
  - Eventual consistency is acceptable for analytics and search features.
  - Data pipelines aggregate and transfer data from the transactional database to the analytical layer for reporting and BI.
  - All proposed tools are open source or free for educational use.

- **High-Level Architecture Overview:**
  The database represents a digital library system designed to manage users, academic resources, and their interactions. At the core of the model, the **User** entity stores essential information such as email, password, name, and status. Each user is associated with a specific **Role**, which defines their permissions in the system (for example, administrator, student, or faculty). In addition, users belong to a **Program**, which connects them to an academic faculty or area of study.

  Users interact with the system mainly through the **Download** entity, which records the resources they access. Each download links a **User** with a **Resource**, registering details such as the time of access. Resources are the central assets of the library, and they contain bibliographic attributes like title, year, publisher, synopsis, language, and identifiers such as ISBN, ISSN, or DOI.

  Resources are connected to several complementary entities. The **Author** entity stores the creators of academic works. Since a resource may have multiple authors and an author may contribute to several resources, this many-to-many relationship is managed through the **Resource_Author** bridge table. Resources may also require legal and usage restrictions, represented by the **License** entity, which indicates whether a resource is externally licensed and includes start and expiration dates. Furthermore,

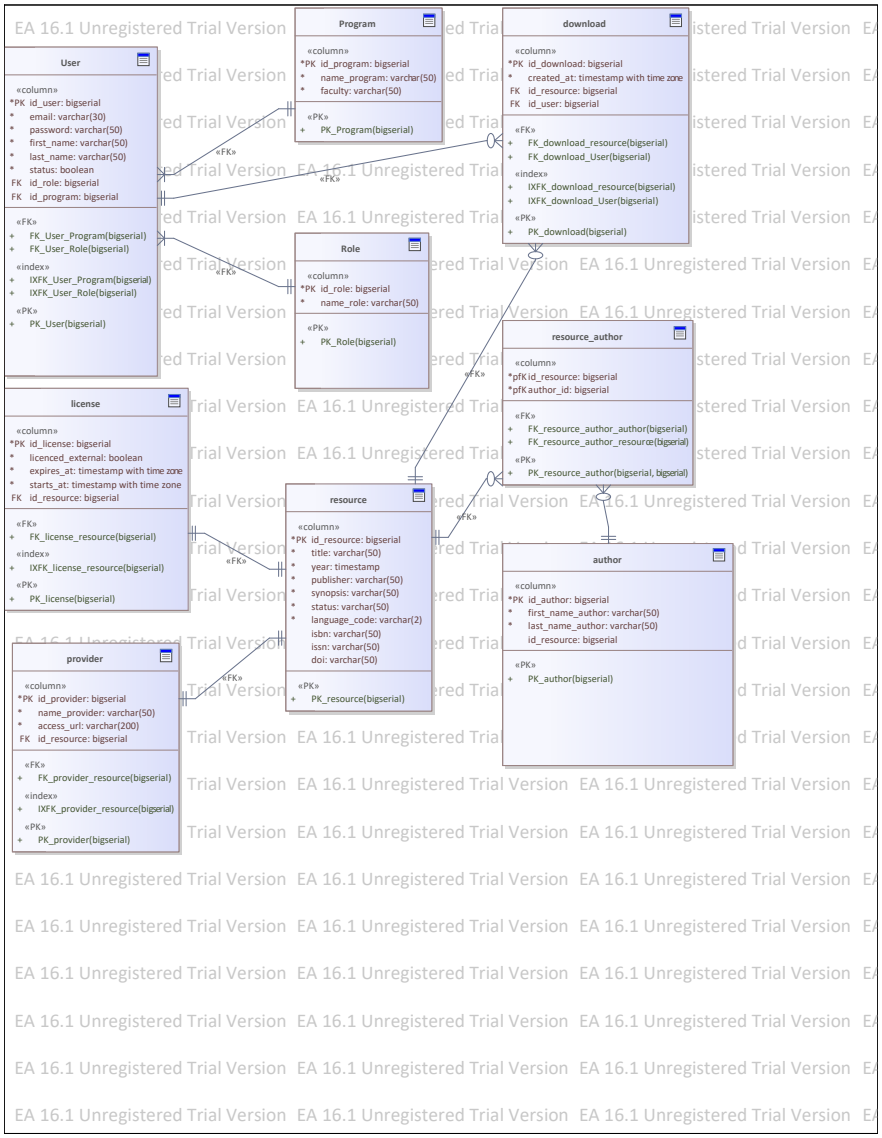

Figure 1: Fisrt versión of Entity Relationship Diagram

resources are linked to **Providers**, which supply the access information, such as provider names and access URLs.

Overall, the model describes a structured environment where users, organized by role and academic program, access digital resources. These resources are enriched with metadata, licensing rules, and provider information, while the system carefully records usage through downloads. The design ensures proper management of academic materials and controlled access based on user roles and licensing conditions.

- **Data Flow and Storage Solutions:**

The data flow in the digital library system is structured across four layers to ensure scalability, availability, and efficient management of both transactional and analytical workloads. Each layer handles a specific set of responsibilities and aligns with appropriate storage technologies.

Presentation Layer. This layer provides the user interface, which may be web or mobile applications. It communicates with backend services using REST APIs or GraphQL, ensuring a seamless interaction between users and the system without embedding business logic in the client.

Application Layer. The application layer manages business logic, request orchestration, and authentication. It controls how data requests are processed and how information flows between the user interface and the database services, acting as the central coordinator of operations.

Domain Layer. This layer encapsulates the core domain entities such as users, resources, downloads, and licenses. It enforces business rules before persisting any data, for example, verifying license validity before a resource can be accessed or downloaded.

Infrastructure Layer. The infrastructure layer integrates several storage solutions. A relational database, PostgreSQL, is used to store structured data such as users, roles, programs, and metadata of resources. Large unstructured files such as PDFs and ePubs are stored in an object storage solution like MinIO. Additionally, OpenSearch is included to provide fast full-text search and filtering across the catalog. Data pipelines are implemented to transfer information from the transactional database into the analytical layer, supporting reporting and business intelligence. Replication, partitioning, and backup strategies are employed to guarantee scalability, high availability, and fault tolerance.

Storage Alignment. The combination of PostgreSQL for structured data, MinIO for digital content storage, and OpenSearch for indexing and discovery ensures that each type of information is managed efficiently. Analytical needs are met through ETL pipelines that move data into a data warehouse or BI environment. This layered architecture guarantees both the integrity of academic records and the performance required for large-scale access.