

1. Introduction

1.1 Project Overview

The purpose of the ByteScribe team, in partnership with the Library of Alexandria (Post-Burn), is to create a digital database (DLoA [Digital Library of Alexandria]) to contain their library, along with providing an easily accessible way to interact with and view information about its contents.

1.2 Scope

The DLoA system will allow both literature enthusiasts and ancient historians to finally peruse and explore the vast shelves that were once so tragically lost in a scathing inferno. The database system will allow various actions such as searching the catalogue, checking a specific item's status, and generating a report based on current trends. The knowledge is no longer lost!

1.3 Glossary

SQL: Structured query language

EECS: Electrical Engineering and Computer Science

KU: University of Kansas

SSH: Secure Shell

RDP: Remote Desktop Protocol

ID: Identification

DLoA: Digital Library of Alexandria

2. Stakeholders

- Staff- employees maintaining state of library
- Donors- individuals and companies that donate time, money, or works for the benefit of the library
- Literature enthusiast- individuals who use the library to satisfy their love of the written word
- Community members- individuals and groups that rely on and support the library
- Researchers- individuals and groups that utilize the library to increase community knowledge

3. Requirements

3.1 Functional Requirements

- Each piece of recovered media shall have identifiable attributes.
- Several categories of members (e.g., regular, students, senior citizens, etc.) may exist.
- Track all borrowing and returning transactions, including timestamps and responsible clients.
- Allow clients to reserve media that are currently on loan.
- Fines owed by each member should be calculated, considering overdue books and a daily fine rate.
- Borrower information should be readily retrievable from the database, such as being able to identify the members who have borrowed the most books in a particular genre in the last year.
- The system must generate a report listing all books due within the next seven days, ensuring that the results are sorted by due date in ascending order. It should also list all members who currently have at least one overdue book, including the titles of the overdue books for each member.
- To analyze borrowing trends, the system must calculate the average number of days books are borrowed for a given genre. Additionally, it must determine the author whose books have been borrowed the most within the past month.
- For financial tracking, the system must generate a report detailing the total fees collected in the last month, with a breakdown by membership type. It must also produce a list of members who have exceeded their borrowing limits, allowing staff to monitor compliance with lending policies.
- The system should be able to determine the most frequently borrowed items by each client type, helping to identify popular resources for different user groups.

Additionally, it must track members who have never returned an item late, recognizing consistently responsible borrowers.

- To measure circulation efficiency, the system must calculate the average time an item stays on loan before being returned. A monthly summary report must also be generated, providing an overview of the total number of items loaned, total fees collected, and the most popular borrowed items.
- For inventory tracking, the system must list all items along with their availability status and last borrowed date. Items that have not been borrowed in the past six months should be highlighted to assist in collection management. It should also generate a report listing all overdue items, including the client responsible and the corresponding late fees.
- The system must summarize the library's revenue from fees, providing a detailed breakdown by membership type and item category. All reports must be generated efficiently, with accurate calculations and appropriate access restrictions to protect sensitive information.

3.2 Data Entities

- Literature Piece: (Book Equivalent)
 - Attributes: MatID, Title, Author, Year (Not Null), Genre, Substrate (Scroll, Stone Tablet, Papyrus, ect.), Availability (Boolean) Due Back Date
- Ancient Mathematics: (Magazine Equivalent)
 - Attributes: MatID, Title, Mathematician, Year (Not Null), Date Proved (\geq Year), Focus (Geometry, Algebra, ect.)
- Work of Art: (Video Storage Equivalent)
 - Attributes: MatID, Artist, Year (Not Null), State of Repair, Restoration Date
- Client:
 - Attributes: AcctID, First Name (Not Null), Last Name, Contact Information (Not Null), Account Tier, Account Age (Not Null)
- Database Administrator:

- Attributes: AcctID, First Name (Not Null), Last Name (Not Null), Contact Information, Activity Status, Age, Permissions

3.3 Non-Functional Requirements

Non-Functional requirements for this database include the following:

- Improved Security (Achieved through spirits of past librarians)
- Retained Work Message (The ego of ancient artists is preserved)
- Ancient Egypt Regulation Compliance

4. Hardware and Software Requirements

The Library of Alexandria database will function by using MariaDB as a SQL server and will operate on the EECS cycle servers provided by KU. There may also be additional software requirements, such as software dependencies associated with MariaDB and SQL.

4.1 Hardware Requirements

Minimum Hardware Requirements:

- Processor: x86_64 or ARM64 CPU (Minimum: 1 GHz, Recommended: 2+ GHz)
- Memory (RAM):
 - Minimum: 2GB
 - Recommended: 4GB+ (for better performance with multiple connections)
- Storage:
 - Minimum: 1GB
 - Recommended: 10GB+
- Network: Access to EECS cycle servers via SSH, with appropriate user privileges

These basic hardware requirements should be easily satisfied and surpassed by the EECS cycle servers. The EECS cycle servers will be accessed via SSH, RDP, or by being on-site. The servers will require KU credentials in order to be accessed.

4.2 Software Requirements

The database will be run on Linux (Ubuntu) using MariaDB. If not already installed, MariaDB will be installed using the following command:

```
sudo apt update && sudo apt install mariadb-server
```

As per KU policy, all KU students will get an account ID and password to access EECS MariaDB (Ver 15.1 Distrib 10.6.18-MariaDB). To connect to the database, these steps must be followed:

1. Log in to one of the EECS Cycle servers
2. Enter the following command:

```
mysql -h mysql.eecs.ku.edu -u userID -p
```

3. Replace userID with your assigned user ID.

Permission to access KU EECS computers will also be needed in order to properly make use of the database and cycle servers. This will be provided by default by using KU credentials.

5. Appendices

No relevant information.