An Integrated Library System that Focuses on Collaboration: Bringing Books and Computers Together

Independent Study Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Bachelor of Arts in Computer Science in the Department of Mathematical and Computational Sciences at The College of Wooster

by
Mae Koger
The College of Wooster
2024

Advised by:

Dr. Thomas Montelione (Mathematical and Computational Sciences)

Abstract

Integrated Library Systems (ILS) are means by which library databases organize metadata relating to the books in their possession, whether physical or digital. Using preexisting systems in conjunction with elements of collaboration, we can create an ILS that focuses on collaboration in an educational environment. This software incorporates folksonomy and other user-generated information into an ILS for use in academic libraries, while keeping in mind user experience. To ensure a good user interface, a usability study was conducted. From this usability study, we can identify changes to make to enhance user experience on the site. Based on the results of our usability study, we made improvements to some signifiers used in the software. Future work on this project includes the addition of login functionality and updating records in the database.

This work is dedicated to Amy and Dave Koger. Thank you for believing in me.

Acknowledgments

To Ragan, thank you. Thank you for your encouragement, you have propelled me forward and made me believe in myself, through everything.

To Emma, thank you for living with me for the past three years. I will never know where I'd be without you.

To EB, thank you for working the night shift with me in the living room. I truly might have gone crazy if we hadn't started working in silence on IS together.

To Dr. Thomas Montelione, thank you for your support and guidance throughout this project.

To all my friends who had to hear me complain about being a Computer Science major, thank you for putting up with me and humoring my complaints.

To the sunrises and sunsets I have missed whilst working on this project, may we meet again amongst birdsong and constellations now that it's over and done with.

Contents

List of Figures

List of Tables

List of Listings

CHAPTER 1

Introduction

Digital libraries deal with knowledge through collection, preservation, organization, and access to resources within physical or electronic-based systems [75]. They serve as communication between authors and readers, data and users, and inventory and librarians through bibliographical data, also known as metadata. These systems are widely used among libraries to relay important data regarding academic literature.

Today's digital libraries are mostly search engines, though the idea of a digital library indicates more supported features than search engines are typically used for [43]. Digital libraries offer support for interacting with materials while search engines do not. These interactions include requesting materials and checking them out. Libraries have access to scholarly resources that are not necessarily available on search engines.

While digital libraries allow for connection between individuals through literature, the existing digital libraries are error-prone and therefore lead to the risk of data loss. This occurs when libraries do not have adequate information regarding the location of a book, whether inside the library or while out on loan [9]. This project will focus on the use of digital libraries as software to facilitate learning, easy retrieval, and the collaboration of library materials in an academic context while minimizing the risk of data loss and enhancing user experience.

This project aims to create a digital library software that is easy to use and widely accessible to users and administrators involved in library processes in an academic setting, whether utilizing its resources as a student, educator, or librarian. This project is based on current digital libraries and their shortcomings along with processes used today in physical libraries—both academic and otherwise.

Choosing a university library setting for this project allows for the use of a narrower dataset in execution. This software will focus on a subset of academic literary materials available for use at a university's library. The materials referenced in this software will be a collection of books discussing literature, cultural, and media studies sourced from Springer Nature [1].

The software will have functionality for all individuals involved in library processes, including students, educators, and librarians. Users should be able to find, tag, and comment on materials, as well as read other users' academic discussions on books through the library.

Connection is essential in digital libraries. The ability to share knowledge through a library system allows for a wider range of impact when providing academic discussion in an online setting. Thus, this software will be cloud-based, which will allow its users to access the system from multiple locations and obtain reference materials within the database, no matter where they access the site from. The works referenced in the software will be those that the library has available and accessible in its physical and digital database through fair use, public domain, and open access.

The purpose of this project is to focus on the use of digital libraries as software to facilitate learning, easy retrieval, and collaboration of materials in an academic context as well as collaboration within this environment. In the creation of this digital library, it is crucial to understand how digital libraries should function in an educational environment as well as explore the features digital libraries currently employ to facilitate learning. This software will allow all users—whether student, librarian, or professor—to effectively communicate with one another. Collaboration in a digital library can include annotations, reviews, or other comments from professionals or students regarding the content at hand in reference to ideas, new discoveries, or other related literature [43].

This project aims to effectively employ elements of user interface (UI) design to allow for intuitive interactions between the software and the user. To ensure the software utilizes an effective UI design, a usability study is necessary. Interviews are to be conducted and will include instructions for the user to follow as well as a questionnaire requesting information about the software's usability based on its interface. Based on survey results, we can implement changes to better the usability of the application.

The software will be constructed using preexisting digital libraries as references for supported features. Previous works highlight important features for digital libraries as well as suggesting new features to include in order to support libraries and their needs.

CHAPTER 2

Library Management Systems

In 1994, the Digital Library Initiative (DLI) was created by the National Science Foundation, establishing digital libraries as a distinct field. The DLI funded over fifty projects focused on interdisciplinary research and collaboration in a digital library setting [26].

Since the 1990s, libraries have engaged in resource sharing with other local libraries [32]. In the early days of resource sharing, libraries exchanged physical material, while today students in academic institutions have access to a greater amount of digital material from local institutions via the connections established by their libraries.

Library management systems (LMS), also known as integrated library systems (ILS), are systematic efforts to organize bibliographical information, or metadata into platforms for user interaction in a library setting. In addition to metadata, ILSs house most of the other necessary records for library administration: "payroll, expenses, purchases, and ... keeping track of various media being checked out by the librarians' patrons," [58]. This application aims to add collaboration to an ILS.

2.1 MARC Records

In the MAchine Readable Cataloging (MARC) Pilot Project in the late 1960s, the Library of Congress tested their format by sending cataloging data in machine-readable form to sixteen libraries, who tested the MARC records on their internal processing applications. The purpose of this project was to make the distribution of library catalog records to libraries easier [12]. MARC records were sent to libraries in the form of magnetic tapes containing “catalog and cross-reference records." These records were encoded on magnetic tapes corresponding to the number on their reference card [13]. In the Pilot Project, MARC records were used to create catalog cards for use in libraries [12]. Upon successful completion of testing, MARC II was created based on the prototype, MARC I, and established as a “permanent part of the Library of Congress" [47].

The main purpose of MARC records was to broaden the availability and accessibility of cataloging information in a machine-readable form for use in an LMS. Prior to this, cataloging in libraries was done by hand on paper. These machine-readable records were intended for use nationally, allowing libraries to manipulate the data to fit their needs [47]. During this time, utilizing processing power for library management was an expensive process [68]. This system would allow for the integration of technology in libraries, making it easier for librarians to keep track of inventory [13].

MARC21, hereafter referred to as MARC, is the most updated version of this project. As used today in LMSs, MARC records contain bibliographical information in a format that is easy for computers to interpret. Most MARC records are meant for administrator or librarian use and therefore are not intended for users. On the front end, MARC records are translated into a user-friendly format for use in integrated library systems. When a user visits a site that hosts bibliographical data, they see MARC records after being processed by the integrated library system for display in the library's site [47].

The original purpose was to create data cards for use internally. Today's use of MARC records is closer to inventory keeping or database organization, meaning that libraries have shifted the meaning and purpose of MARC records over time. With the shift in purpose of MARC records since their creation comes a few problems. Firstly, they can only store text-based information. Thus, the data stored cannot be "reviews, indexes in their original form, and image and sound files." Secondly, there is no method by which records can be linked to one another utilizing MARC records [12].

2.2 Problems with Library Management Systems

Since the creation of MARC records and the introduction of computing in library processes, library functions have outgrown LMSs. In academic libraries, electronic resources have become more widely used than physical resources. The incorporation of resources available online into these systems created in the late 90s suggests the market for an updated system to handle today's library functions [70].

The incorporation of new technologies in libraries will require the employment of librarians who know how to use them. Syeda Batool's research in Pakistan libraries focused on determining the constraints in gaining IT skills [16]. Through interviews, Batool found that the use of refresher courses for librarians focusing on IT skills as well as the adoption of technical evaluation within libraries was useful.

A few challenges LMSs face include ease of use, error reduction, and efficiency. In their Ph.D. thesis, Tobi Adebesin proposed a system that will solve these problems of existing library systems. These problems include the introduction of errors and the lack of timeliness when manually entering information into a management system.

Adebsin's software effectively reduced the tedium in manually creating entries in library systems by reducing manual bibliographical tracking when recording the location of books within the library. This project also aimed to be used as electronic storage and inventory tracking information for use in academic libraries [9].

2.3 Notable Projects and Software

In 1945, Vannevar Bush proposed a digital library called a memex, a device that would store data relating to one's library for personal use, operated using a computing device, and was similar in function to today's digital libraries. This device was intended to work in a similar manner to the human mind to organize information. To do this, Bush suggested the use of a recommender system [19].

Another initial concept that served as the inspiration for digital libraries came from J.C.R. Licklider. After hearing of Vannevar's proposal, Licklider recognized that it might be possible to create such an automated library system with the increased power of computers [20].

Generic library management systems began to appear in the early 2000s and were similar to repository management systems [20]. Today's digital library software allows for this type of repository management, along with additional features including searching, filtering, and collaboration [43].

A few other historical examples of digital libraries to note are *arXiv*, Electronic Thesis and Electronic Dissertations repositories (*ETDs*), and An Integrated Art Analysis and Navigation Environment (*ARTISTE*) [20].

2.3.1 Fedora

First developed in 2001 at the University of Virginia [64], the Flexible Extensible Digital Object Repository Architecture (Fedora) is an open-source framework for the creation of information network overlays to “manag[e] and deliver complex digital objects" in a repository. Fedora allows for the use of relationships between objects in the repository. The architecture supports the use of complex objects such as “documents, images, electronic books, multimedia learning objects, data sets, [and] computer programs," [42].

The Fedora architecture supports version control. When an object is changed, the software creates a new version upon modification and saves it along with a change log, which stores information regarding the nature of the modification to the object. This includes “who, what, when, where, and why an object was changed," [42]. This allows users to recover previous versions of records in case a mistake is made.

2.3.2 Greenstone

An open-source software that facilitates the creation and presentation of information collections [76], Greenstone, was developed in 1995. Initially named the New Zealand Digital Library project [73], the software allows users to create digital libraries consisting of information collections [76]. This software has functionality for multiple login types for workers in a library, depending on the user’s job title [74]. Rather than focusing on searching, collaboration, or checking books in or out, this software is used for the cultivation of library collections [74]. Some challenges with the Greenstone toolkit include the low level of customization and the use of immutable files generated during the building process [14].

Since the creation of Greenstone, there have been several attempts to improve upon its features. An architecture implementation created by Bainbridge and others aimed to improve customization and usability in the building process of collections using the Greenstone toolkit. Greenstone3, created by David Bainbridge et al., introduced collaboration to the toolkit. These collaborations include highlights within documents and tagging contributed by users in an effort to make it easier for them to quickly scan the documents for pertinent information. Greenstone3 utilizes Cascading Style Sheets (CSS) to preserve the look of previous Greenstone projects [14].

2.3.3 Project Gutenberg

Started in 1971, Project Gutenberg is a site where users can download free digital copies of books that have expired copyrights. Stroube explains the importance of having digital copies of literary materials for preservation, which emphasizes the usefulness of digital libraries that store digital books [65]. The site hosts electronic texts (e-texts), which are different from electronic books (e-books). E-books require the use of reader programs such as Glassbook Reader or netLibrary eBook Reader, which were not available on Macintosh computers for quite some time. Project Gutenberg's e-texts are text files containing ASCII (American Standard Code for Information Exchange) characters and can be read on nearly any computer, reducing the requirement for extra hardware and therefore broadening the accessibility of their texts [15], since any web browser can process and display files containing ASCII characters.

2.4 Collaboration in Library Management Systems

Digital libraries are defined by their ability to support collaboration between users [20] by collecting their readers' opinions. Yao's ideas about the applications of computer science in libraries emphasize the use of collaboration as a tool for libraries in an academic setting [77].

ILSs are different from search algorithms. Search algorithms only have the capacity for searching and sorting through data while ILSs allow for searching, sorting, and collaborating. Since advances in computer science have made a digital representation of literature widely available through digital libraries, they have evolved into “complex networked systems able to support communication and collaboration" among those who use them [20]. In some contexts, it may be useful for an ILS to have collaborative features [43]. For example, research in a scientific context through an ILS may benefit from collaboration. This would allow verified users to discuss the impact of a given study, other studies that are similar, new discoveries that impact the given study, or any other pertinent information they want to discuss with colleagues. Collaboration in this context consists of input from users via annotations and corrections. Further examples of collaboration in an ILS include keywords, ratings, and connections to other documents [75].

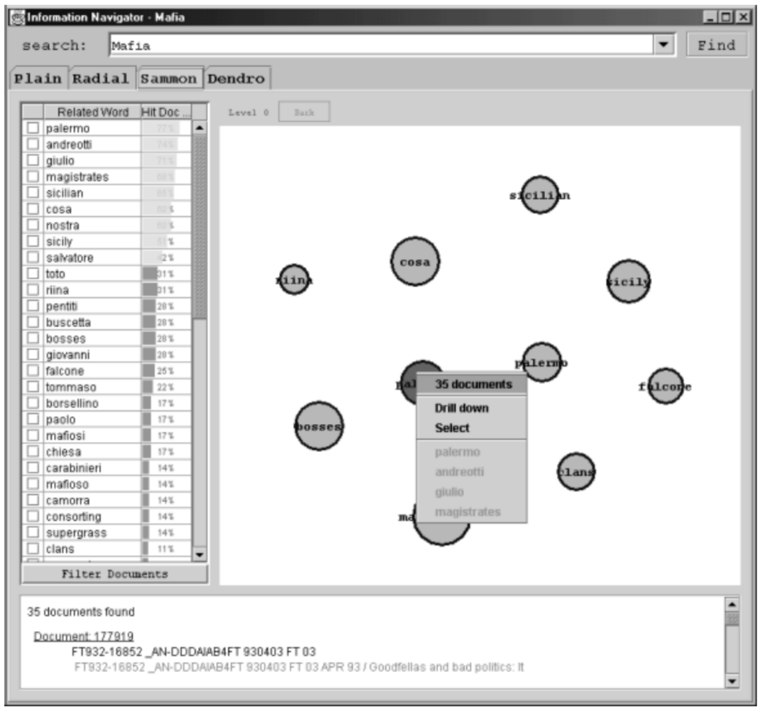
There have been recent trends in user-provided collaboration in ILSs as well as increased library, archival, and museum (LAM) interdisciplinary collaboration. An example of LAM collaboration is the Smithsonian’s The Commons, a photostream posted on Flickr by the Smithsonian that allows users to comment on and add tags to photos in collections. Using online-based user-created collaboration spaces allows for interdisciplinary academic conversations on topics that otherwise may not have occurred [52], enabling library materials to have an increased audience when users comment on and link documents to one another.

2.5 Folksonomy

Folksonomy is a collaborative effort to create and assign tags to documents from user input. The term was coined by Thomas Vander Wall in 2004 as an “idea of socially constructed classification," posted in a blog by Gene Smith [55].

The idea of user-generated tags introduces a few issues. One of note is the lack of precision involved when users create their own tags [55]. This results from potential misunderstandings of the purpose of tagging. In some cases, user-generated tags may be collected independently of one another, meaning that users cannot see tags created by others. This might create an opportunity for a wider variety of tags on documents [60]. This allows for the possibility of multiple existing tags that mean the same thing. For example, the tags *tv* and *television* both mean the same thing but are not evaluated as the same when performing searches. Authority control can be used in the creation of tags by including a list of system-generated tags for users to select or using Natural Language Processing (NLP) to allow or disallow tags based on part of speech or other constraints and guidelines [55] put in place by administrators. Supporting functionality for private and public tags can allow users to create a database of documents for personal use as well as contribute to socially generated categorization.

When taking user input for collaborations, it is important to utilize authority control to ensure the comment follows community guidelines. We also want to ensure the document's MARC record is changed only by those authorized to do so. This can be done by using relational databases, described in Section 2.8.2.2.



	Related Word	Hit Doc ...
☐	palermo	77%
☐	andreotti	74%
☐	giulio	71%
☐	magistrates	68%
☐	sicilian	65%
☐	cosa	60%
☐	nostra	60%
☐	sicily	51%
☐	salvatore	42%
☐	toto	31%
☐	riina	31%
☐	pentiti	28%
☐	buscetta	28%
☐	bosses	28%
☐	giovanni	28%
☐	falcone	25%
☐	tommaso	22%
☐	borsellino	17%
☐	paolo	17%
☐	mafiosi	17%
☐	chiesa	17%
☐	carabinieri	14%
☐	mafioso	14%
☐	camorra	14%
☐	consorting	14%
☐	supergrass	14%
☐	clans	11%

35 documents found

Document: 177919
FT932-16852 _AN-DDDAIAB4FT 930403 FT 03
FT932-16852 _AN-DDDAIAB4FT 930403 FT 03 APR 93 / Goodfellas and bad politics: It

Figure 2.1: Graphical Cluster-Based Search Interface [21].

Tag-based searching allows users to have an easier time searching for items in a database. This can be due to the increased mental energy required to recall relevant search terms over selecting a term from a list [60]. By using this method of searching, a graphical cluster-based interface in searching might better aid users in finding appropriate search terms for their queries [21]. Figure 2.1 shows an example of a cluster-based search to find documents in a database.

2.6 Copyright Law

“In copyright law, possession of a copy of a document certainly does not constitute ownership," [75]. Thus, libraries assisting individuals with access to material must have the legal ability to do so by obtaining the rights to that material with the ability to loan the material out. The fair use principle allows for limited copying and distribution of copyrighted documents to uphold “society’s access to ideas and innovation," [45]. Through fair use, public domain, and open access, libraries can provide literature, both physical and digital, to their users at no cost.

2.6.1 Fair Use

The fair use doctrine allows the use of copyrighted works “to preserve ... society’s access to ideas and innovation” [45], permitting libraries to distribute works that are copyrighted. There are five factors used to determine if a work can be considered fair use:

1. What is the work being used for?
2. Is the work published or unpublished?
3. What percentage of the total work is used?
4. Will the free use of the work cause the author to lose money?
5. Are the potential uses of the work considered “honorable?" [44, 45].

Within academic libraries, fair use is crucial in the “noncommercial purposes of research, course work, scholarship and teaching" [56] when dealing with copyrighted works, and are thus available to the public at no cost.

2.6.2 Public Domain

Works created in and after 1978 have a copyright length of the author’s lifetime plus seventy years [56]. Documents that have outlasted their copyright or were created before 1923 [44] enter the public domain upon expiry. Project Gutenberg, described in Section 2.3.3, aims to provide a digital library of works that are in the public domain. These works do not require permission to be distributed [65], and are thus available to the public at no cost. Thus, today there are far more open access articles that can be included in library databases.

2.6.3 Open Access

Articles and documents that are published and freely accessible online are considered open-access materials. This is a result of the increased availability of low-cost distribution of documents on the internet. Prior to this, journals were distributed to subscribers in the mail, meaning they had to have a wide enough fan base to justify the expensive printing and distributing costs. With open access, journals can publish articles with niche topics without having to worry about justifying costs to distribute content [41].

2.7 Open Source Library Management Software

A database of bibliographical records is required to make and run any ILS, which can become expensive when coupled with keeping up with technological advancements. There are many applications that use MARC records for the display of information, some of which are Evergreen and Koha, front-end applications used for checking out and browsing books on both the user and librarian logins. Both Evergreen and Koha are open-source LMSs and thus are available for libraries to use and modify to suit their needs, as necessary.

Open-source software is available to everyone without cost [58], and allows libraries to have access to systems that more closely fit their needs. Libraries are able to modify the programs based on their needs and redistribute them without royalties [61]. This introduces challenges for libraries who wish to modify the source code to add a supported feature for their site. This requires the employment of an expert in the field who can successfully do this since most open-source software does not handle such things [58].

Koha claims to have been the first open-source LMS and has the ability to support all library functions, such as “circulation, cataloging, acquisitions, and serials," [61, 66].

2.8 Data Management

When dealing with data, it is important to consider how the data will be stored, and by whom it might be modified. There are several ways of dealing with data storage and processing. Ideas to consider when working with data are cloud computing, database selection, and relational databases.

2.8.1 Cloud Computing

Cloud computing, introduced in 2006 by Eric Schmidt [57], utilizes the storage of information in data centers instead of locally to save on expensive hardware and software required to store data on-premises [8]. On-premises infrastructure is expensive and less scalable than cloud computing [10]. The information stored ‘in the cloud’ (infrastructure, platform, or software) can be accessed anywhere with an internet connection [36], assuming the user has access to the site with their credentials.

A data center can store information for multiple clients, pooling their resources to save space. Providers deliver data to their clients over the internet [24]. This allows for “cheap and easy access to externalized IT resources," [63].

As illustrated in Figure 2.2, cloud users access resources remotely via a network connection to the data center which stores the information requested [11].

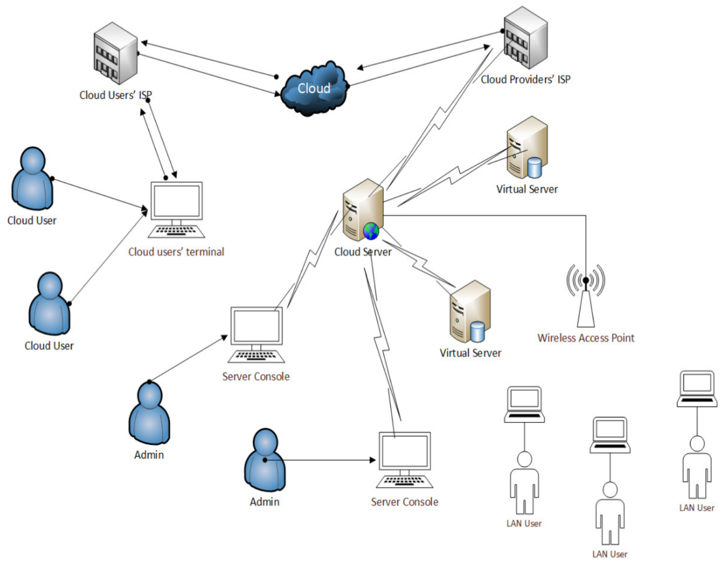


Figure 2.2: Cloud Computing Architecture [11].

Two types of cloud computing to note are those that provide computing instances and those that support data-intensive computations. Cloud computing that provides computing instances supply software or a platform as a service. Data-intensive computations involve the management of large amounts of data [27].

With cloud computing comes sustainability risks as well as intellectual property risks [67] due to the data being stored with a third party. Thus, cloud computing should be the preferred method of storage for those whose intended audience is the general public [8].

Another risk introduced by cloud computing is the risk of clients' applications compromising hardware. If this hardware is shared with other clients, this might cause data loss or damage [27].

2.8.2 Database

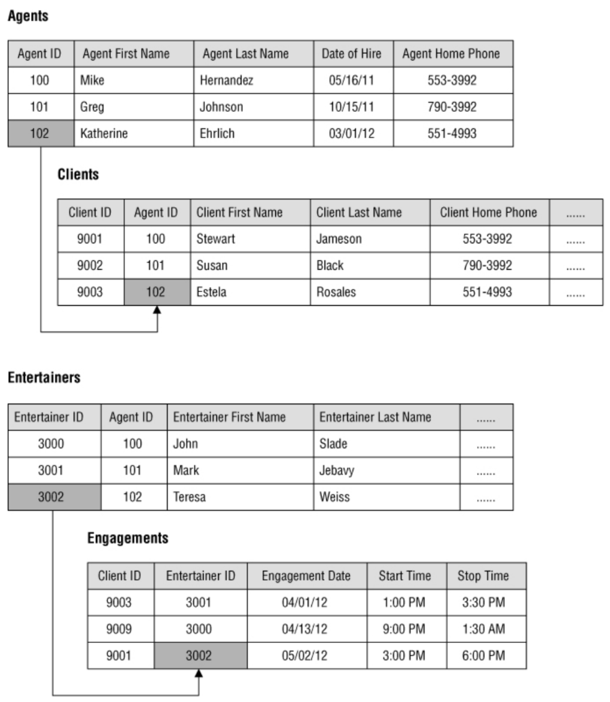
When selecting a database for an ILS, it is important to consider whether a relational or non-relational database should be used. Relational databases utilize multiple tables that use primary and foreign keys to connect corresponding information while nonrelational databases store all information in one table or another form of data storage in this manner.

2.8.2.1 Non-Relational Databases

Non-relational databases, also known as NoSQL, "is a set of concepts that allows the rapid and efficient processing of data sets with a focus on performance, reliability, and agility," [46]. NoSQL databases allow for the use of key-value pairs, graph data stores, and document stores as data formats [25]. Examples of this include "JSON, graphs, [and] key-value hash," [22].

2.8.2.2 Relational Databases

Introduced in 1970 by Dr. Edgar Codd, relational databases extended from mathematical set theory. Relational databases are those that utilize multiple tables associated with one another via unique primary and foreign keys associated with entries in the table [30]. Primary keys must be unique so table entries can be associated with the correct values and attributes.



Agents

Agent ID	Agent First Name	Agent Last Name	Date of Hire	Agent Home Phone
100	Mike	Hernandez	05/16/11	553-3992
101	Greg	Johnson	10/15/11	790-3992
102	Katherine	Ehrlich	03/01/12	551-4993

Clients

Client ID	Agent ID	Client First Name	Client Last Name	Client Home Phone	......
9001	100	Stewart	Jameson	553-3992	......
9002	101	Susan	Black	790-3992	......
9003	102	Estela	Rosales	551-4993	......

Entertainers

Entertainer ID	Agent ID	Entertainer First Name	Entertainer Last Name	......
3000	100	John	Slade	......
3001	101	Mark	Jebavy	......
3002	102	Teresa	Weiss	......

Engagements

Client ID	Entertainer ID	Engagement Date	Start Time	Stop Time
9003	3001	04/01/12	1:00 PM	3:30 PM
9009	3000	04/13/12	9:00 PM	1:30 AM
9001	3002	05/02/12	3:00 PM	6:00 PM

Figure 2.3: Example of a relational database table [31].

Figure 2.3 shows an example of a relational database. In this example, there are four tables that reference the primary keys of other tables. The table entitled 'Entertainers' references elements in the 'Engagements' table as well as the 'Agents' table via their unique identification number. This allows us to store more information than possible in a NoSQL table as we can have multiple entries with the same foreign key in a table, allowing for one-to-many relationships within the database.

2.8.3 Structured Query Language

Structured Query Language (SQL) is a text-based language that provides the ability to manipulate data when using relational databases [35].

MySQL is a management system that allows its users to read and manipulate relational databases [2]. Its advantages include web applications, open-source support, low overhead, and stability [50].

2.9 User Interface

When a user interacts with an application, the way in which they process visual information from the User Interface (UI) has an impact on their interactions with and understanding of the software. Considering how a user might interpret visual elements is crucial in enhancing user experience in an application.

Before a user can interact with an application or a website, they must first interpret the things they can do with the application: the affordances. Affordances “define what actions are possible," while signifiers indicate to a user the ways in which they can interact with something [51]. It is imperative to include the correct affordances and signifiers to communicate to the user how to interact with the application. For example, the placement and type of a door handle can indicate whether it should be pushed or pulled by the user. By communicating clear goals and following guidelines of interface design, users should be able to use an application without having to read a guidebook or manual [23].

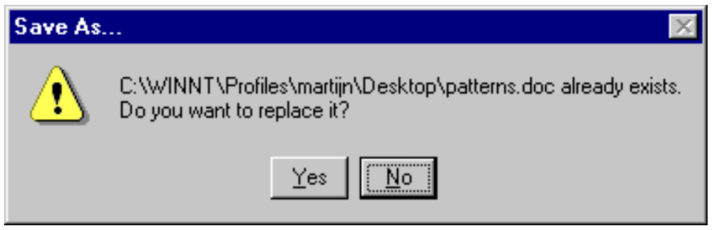
Name	**The Shield**
Problem	**The user may accidentally select a function that has irreversible (side) effects.**
Usability Principle	Error Management
Context	The user needs to be protected against unintended or accidental actions that have irreversible (side) effects. The (side) effects may lead to unsafe or highly undesired situations. For example the unintended deletion or overwriting of files. Do not use for actions that are reversible.
Forces	- The user is striving for speed while trying to avoid mistakes. - The severity of the (side) effects.
Solutions	**Protect the user by inserting a shield.** Add an extra protection layer to the function to protect the user from making mistakes. The user is asked to confirm her intent with the default answer being the safe option.
Examples	 Save As... C:\WINNT\Profiles\martijn\Desktop\patterns.doc already exists. Do you want to replace it? A copy of the file already exists at the specified location. Overwriting it will result in loss of the copy. The default is "No" so that the speedy user has to take the effort of saying "Yes".
Usability Impact	Increased safety, less errors and higher satisfaction. However, it requires extra user action which leads to lower performance time.
Rationale	The extra layer causes the user to require 2 repetitive mistakes instead of 1. The safe default decreases the chances for a second mistake.
Known Uses	Microsoft Explorer, Apple Finder

Figure 2.4: Example of a UI design guideline [69].

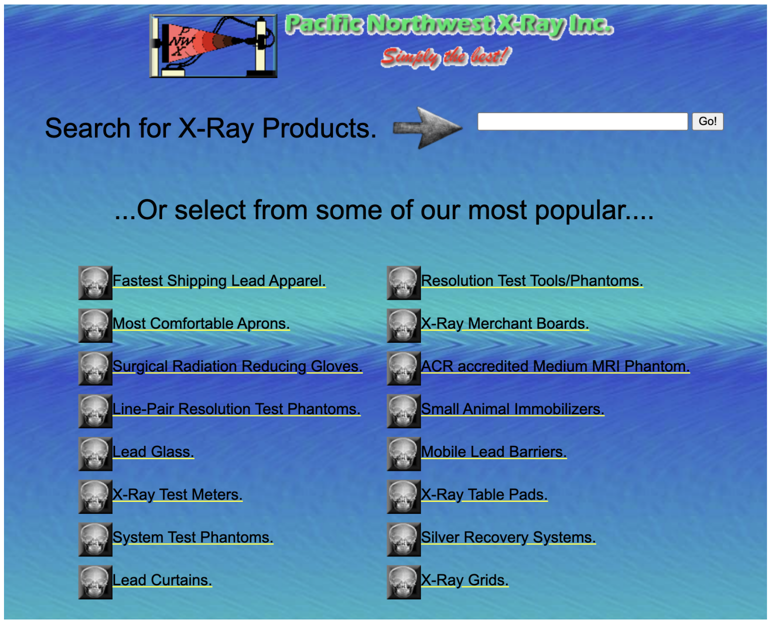
Guidelines, or patterns, exist in UI design to aid in the creation of applications. Figure 2.4 displays an example of a pattern to follow in a circumstance in which a function with irreversible effects is called [69]. Many principles of UI design can be used to effectively create an application, including, but not limited to:

- Aesthetics
- Consistency
- Predictability
- Recovery

When creating a software that has a user interface, the principles of UI design must be considered. While there are several principles in addition to those mentioned, aesthetics, consistency, predictability, and recovery are the focus of many applications. These UI design principles are further described in the following sections.

2.9.1 Aesthetics

Aesthetics refers to the composition of a visual element. If a UI is aesthetically pleasing, it will attract the user's attention and clearly convey its message [18].



Search for X-Ray Products.

...Or select from some of our most popular....

Fastest Shipping Lead Apparel.	Resolution Test Tools/Phantoms.
Most Comfortable Aprons.	X-Ray Merchant Boards.
Surgical Radiation Reducing Gloves.	ACR accredited Medium MRI Phantom.
Line-Pair Resolution Test Phantoms.	Small Animal Immobilizers.
Lead Glass.	Mobile Lead Barriers.
X-Ray Test Meters.	X-Ray Table Pads.
System Test Phantoms.	Silver Recovery Systems.
Lead Curtains.	X-Ray Grids.

Figure 2.5: Bad UI Design [3].

Figure 2.5 shows an example of bad aesthetics in its UI. The background image chosen for the site makes it difficult to read the text because of the colors and patterns used, which can overwhelm the viewer. On the site, the banner flashes between red and black every second, which can be distracting for users. These elements included in the site might influence the user to navigate away from the page, which is not the goal in creating an application. The use of contrasting colors should be considered when designing an application to decrease the risk of usability issues relating to color [37].

Figure 2.6 displays an image of good aesthetics in UI design as it features colors that contrast with one another and images that do not distract the user from the purpose of the software.

2.9.2 Consistency

When a user interacts with a new interface on an application, it takes time for them to learn how to use it. This can be time-consuming and frustrating. To avoid this, we can follow the principle of consistency. We can use similar interfaces to reduce the time it takes for the user to process and learn how to use the new interface [18]. Consistency can be incorporated in applications by setting a standard for repeatable things in the site, such as displaying a search or login button in the same location on each page, with the same styling [37]. Figure 2.6 employs consistency by utilizing three scrollable boxes, each with selectable options that highlight when a choice is made. It is important to note, though, that the box in the “ColorSync Profile" section has a scroll bar while the other two boxes do not. To further employ consistency within the software, it is suggested that the scroll bar is either added to the other two boxes or removed altogether.

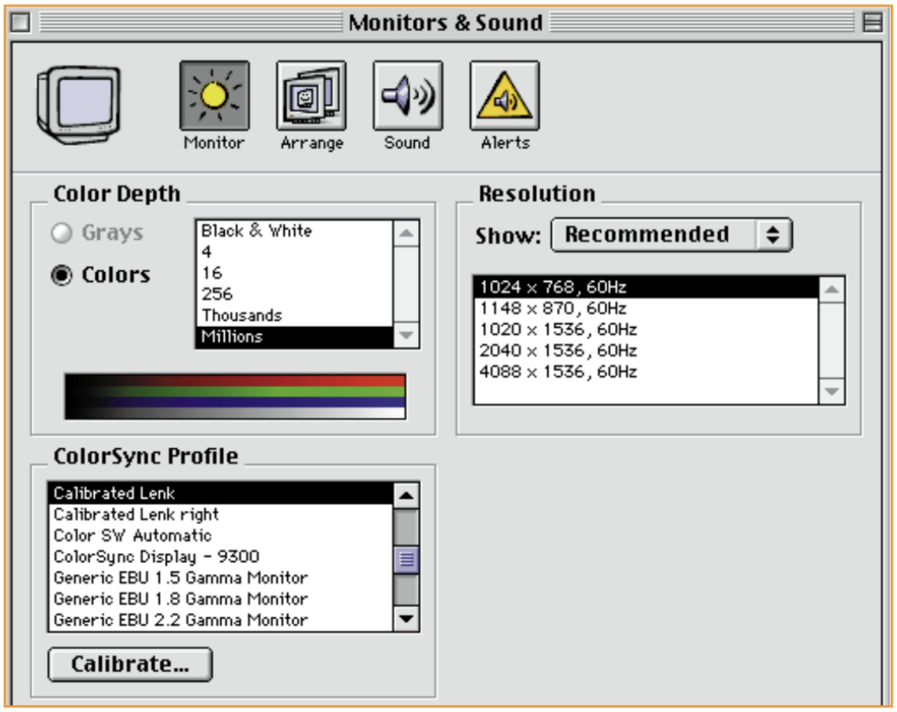


Figure 2.6: Good UI Design [38].

2.9.3 Predictability

User interfaces should utilize interactions that are traditionally used in UI design to reduce the need for users to learn how to use the application [18]. We can use text *displayed like this* to indicate its use as a hyperlink, which allows the user to assume that, upon clicking, they will be taken to another page. Another example of predictability in UI design is the use of a $\boxed{\text{submit}}$ button, which indicates that the user's entered information will be saved upon clicking.

2.9.4 Recovery

When there is an opportunity for a user to make a mistake or lose unsaved changes, it is crucial for the user to be able to recover what they might have lost. This can be done by displaying "are you sure?" messages when a user is about to do something they cannot undo, as well as the incorporation of a back button [18] in case the user is not ready to make a permanent change. These are also known as lock-ins [51]. Figure 2.4 shows an example of a UI guideline for recovery.

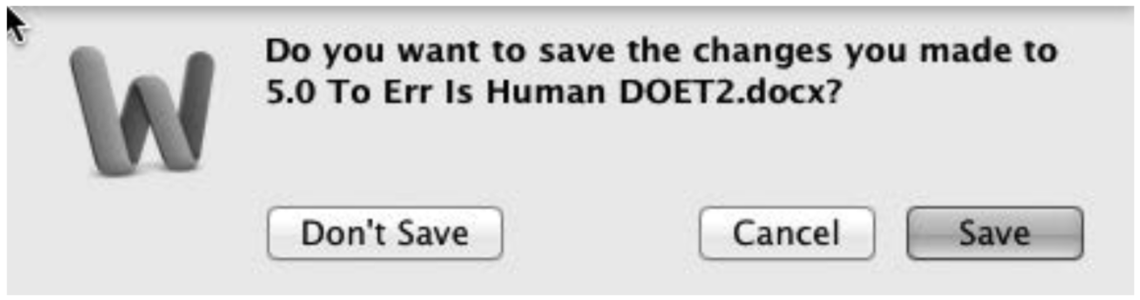

Do you want to save the changes you made to 5.0 To Err Is Human DOET2.docx?

Figure 2.7: Lock-in in Microsoft Word [51].

Figure 2.7 shows a system message allowing the user to go back before making an irreversible change. This popup allows the user a chance to save their document before closing the window.

2.10 Searching

To display query results in an application, a few things to consider include how many results should show on one page, how to match search terms with elements in the database, how to display search results, and what to do if there are no results from a search. The following sections describe these topics.

2.10.1 Paging

When a user makes a query on the search page, depending on the specificity of their search, multiple results will show. Most search algorithm pages handle this by paging. The standard number of items in a search engine results page is ten, though some sites use a different amount or even allow the user to decide [39]. A study done by Michael Bernard et al. found that, while more users preferred having more pages with fewer items per page, the users who had more items per page found documents in a faster manner [17].

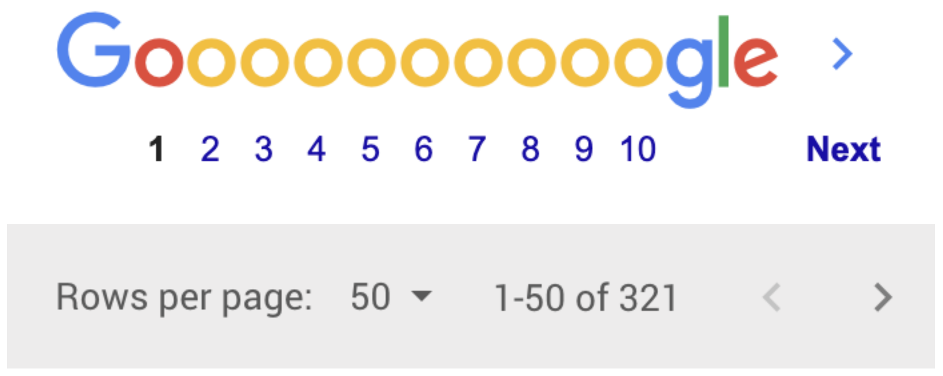


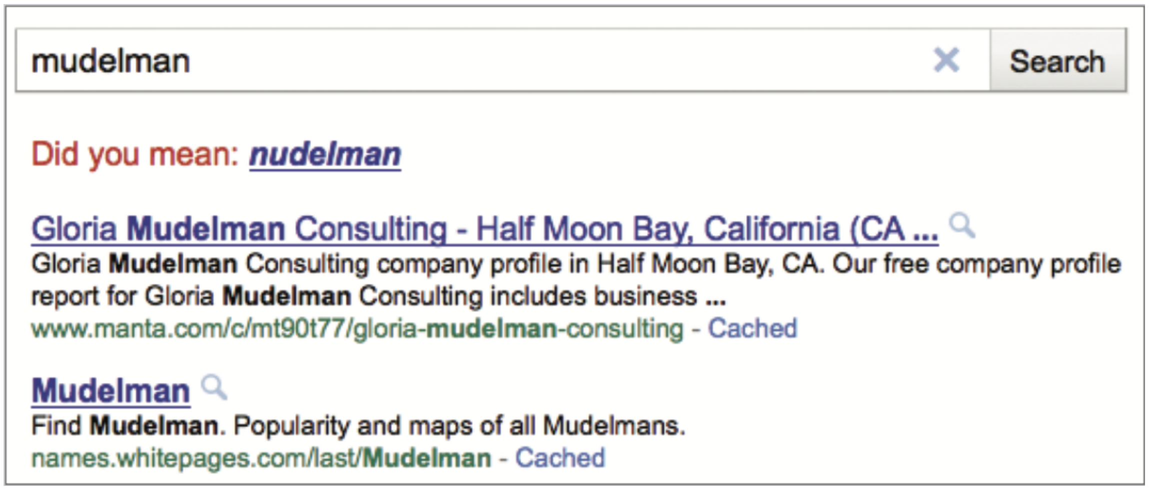
Figure 2.8: Paging section in Google (top) and Presence (bottom) [4, 5].

Some sites allow their users to decide how many results to show in their search results page. Figure 2.8 shows paging sections from two sites. Google's paging does not allow users to change the number of results per page. Presence's search results page allows the user to specify the number of results that are shown per page, along with arrows to navigate between results pages. The options on most sites for the number of results to display are ten, twenty-five, fifty, and one hundred [39].

2.10.2 Stemming

If a user makes a query, it is likely that their search terms exist in the database in another form. For example, if a user searches for "poem" in the title field and there exists an article entitled "The Poetry of Mary Robinson" in the database, this file is relevant to the user's search and therefore should be displayed in the search results. This can be done through stemming, a process that converts a word to its original form, without prefixes or suffixes. When searching, words that have the same root will be considered equal, regardless of their prefixes or suffixes [72]. In the search described above, with stemming, a search for "poem" would yield results that contain the words “poetry," “poems," and “poet" since the words all have the same root.

2.11 Displaying Results


mudelman

Did you mean: ***nudelman***

Gloria **Mudelman** Consulting - Half Moon Bay, California (CA ...
Gloria **Mudelman** Consulting company profile in Half Moon Bay, CA. Our free company profile report for Gloria **Mudelman** Consulting includes business ...
www.manta.com/c/mt90t77/gloria-**mudelman**-consulting - Cached

Mudelman
Find **Mudelman**. Popularity and maps of all Mudelmans.
names.whitepages.com/last/**Mudelman** - Cached

Figure 2.9: Google’s No Results Page resulting from a misspelled search [53].

In the case that a user’s query does not yield any results, it is important to display something to indicate this to the user. Possibilities include partial search results, displayed in order of relevance, as well as an indicator stating that no results were found backed by a “did you mean...?" statement which suggests other searches based on the initial search made by the user using natural language processing. Figure 2.9 shows Google’s ‘No Results Page’ [53].

CHAPTER 3

Methodology

This software aims to create an Integrated Library System (ILS) for the use of searching, surfing, and collaborating on literature materials in an academic setting, allowing users to interact with one another. This encourages academic conversation about papers, articles, and books stored in the ILS. This software is intended for use in an academic library for collaboration among students and faculty.

3.1 Implementation

The application allows users to browse the database by searching and retrieving information from the book details page. The details page also allows users to write comments and tags to the database.

For this application, we will be using Python3, hereafter referred to as Python. To display this site we will be using the Python library Flask, which allows us to host a development server locally, and then deploy to the cloud. With the Flask library in Python, we can establish navigation between HTML pages using the URL (web address) entered by the user or supplied with an internal link.

We will be using app routes, provided by the Python library Flask, in our **app.py** file to render the HTML files used in the application. This is where we make the queries to our SQL database, which displays bibliographical information sourced from SpringerNature [1]. This is further described in Appendix A. The **app.py** file, shown in part in Listing 3.1, is compiled to start the server by entering in the terminal the command `python3 app.py` or `flask run`. This allows the user to connect to the software on their device. This Python file establishes a connection that runs on the URL http://localhost:5000.

```
# application landing page
@app.route('/', methods=['GET'])
def index():
    return render_template("home.html")

# home page
@app.route('/home.html', methods=['GET',"POST"])
def home():
    if request.method == "POST":
        . . .

    # if method == 'GET' render home page
    return render_template("home.html")
```

Listing 3.1: App route for **home.html**.

This site was formatted using HTML (HyperText Markup Language) pages along with Cascading Style Sheets (CSS). CSS allows us to reference established styles rather than repetitively using style attributes in tags. Listing 3.2 shows a portion of the CSS styles used in our HTML documents. Having these CSS styles declutters the HTML page and decreases the repetition of adding style tags to each element. This also allows us to use consistent styles throughout the site.

```
/* styling the heading portion of the site */
.header{
    background-color: #d0d0d0;
    padding: 20px;
    font-size: 10px;
    text-align: center;
    color:rgb(0, 0, 0);
}

/* used for book info in the book details page */
```

```
.float-left{
    float: left;
    padding: 5px;
}

/* used for the comment and tag section of the book info
   page */
.float-right{
    float: right;
    padding: 5px;
}

/* styling for submit buttons */
.button {
    border: none;
    border-radius: 25px;
    color: rgb(0, 0, 0);
    background-color: #ffffff;
    text-align: center;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}
```

Listing 3.2: CSS for use in HTML.

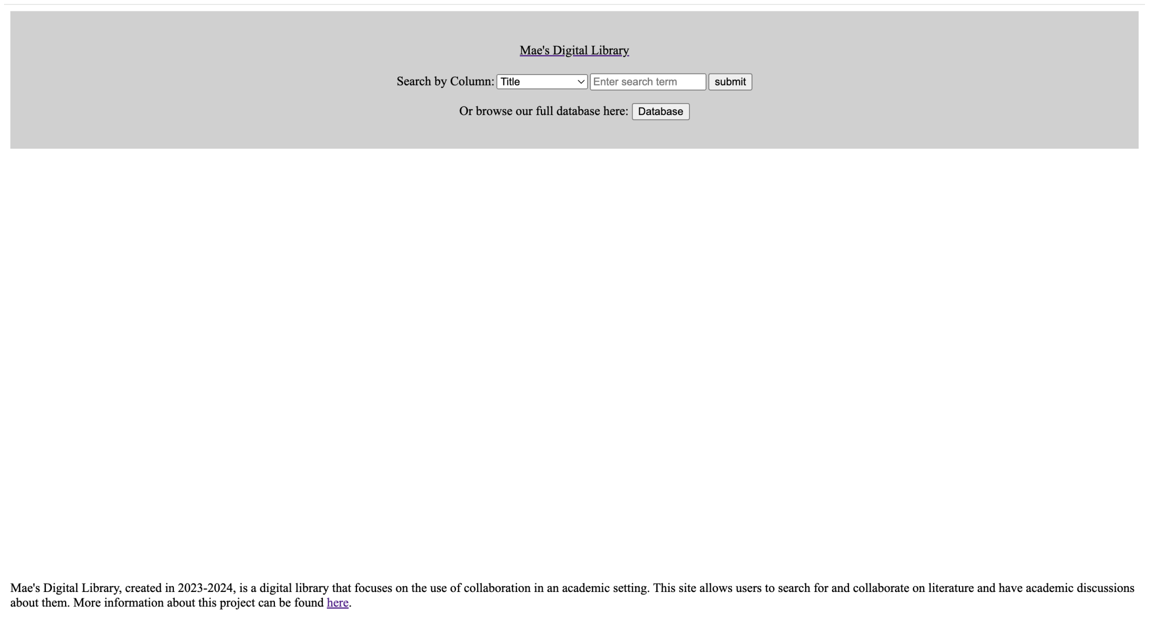
These styles are referenced in the HTML file, as shown in Listing 3.3. The CSS file is referenced in the <**head**> tag on line 6 and specified within tags throughout the body to indicate a specified document to refer to for styling when rendering the content on the user's browser. The class *header* is referenced on line 10 of the HTML file in Listing 3.3, which points to the formatting specifications listed in the CSS file under the same class name.

```
1  <html>
2      <head>
3          <meta charset="UTF-8">
4          <meta name="viewport" content="width=device-width,
               ␣initial-scale=1.0">
5          <title>Home</title>
6          <link rel="stylesheet" href="static/site-style.css
               ">
7      </head>
```

```

    <body>
        <div class="header">
            <a href="home.html">
                <div class="header">Mae's Digital Library</div>
            </a>
        </div>
        ...
    </body>
</html>
```

Listing 3.3: Using CSS in HTML.

The code shown in Listing 3.3 is part of the HTML file that displays the home page of the site, shown in Figure 3.1.



Mae's Digital Library, created in 2023-2024, is a digital library that focuses on the use of collaboration in an academic setting. This site allows users to search for and collaborate on literature and have academic discussions about them. More information about this project can be found here.

Figure 3.1: Application home page.

The landing page for the site, designated with the app routes shown previously in Listing 3.1, is **home.html**. On this page, shown in Figure 3.1, the user can search the database as well as see some information about the site.

The app route attributed to **home.html** allows for *GET* and *POST* methods. The *GET* method, shown in Listing 3.1 gets information from a specified source, in this case, it allows us to read information on the site, while the *POST* method allows the user to input information to be used, which is then processed on the server. In **home.html**, we use the *POST* method to read information entered by the user in the search container. The *POST* method for this app route reads information from the form in **home.html**, shown in Listing 3.5, and uses it to make a search query, returning the results in **search.html**.

```
1  <form action="#" method="post">
2      <label for="searchColumn" style="font-size:␣medium;">
           Search by Column:</label>
3      <select name="searchColumn"id="searchColumn">
4          <option value="BookTitle">Title</option>
5          <option value="ElectronicISBN">Electronic ISBN</
               option>
6          <option value="Author">Author</option>
7          <option value="CopyrightYear">Copyright Year</
               option>
8      </select>
9      <input name="searchTerm"type="text" id="searchTerm"
           placeholder="Enter␣search␣term">
10     <input type="submit" value="submit"/></p>
11 </form>
```

Listing 3.4: *POST* request information retrieval in the home app route.

This form asks the user for input. The search column is collected using the drop-down menu populated from lines 3 through 8, while the search term is collected on line 9 using text box input. Figure 3.2 shows the expanded drop-down menu for selecting a search column when searching in the database.

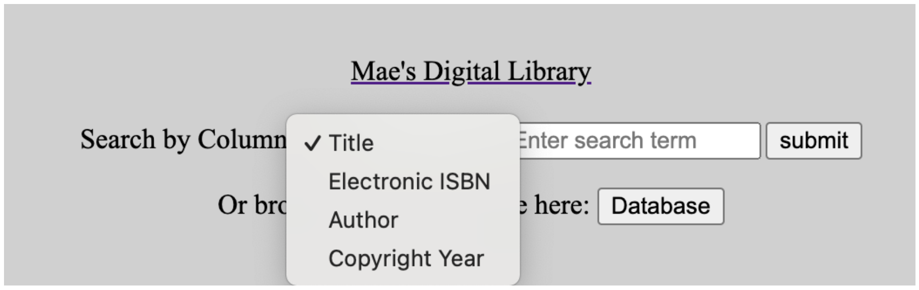
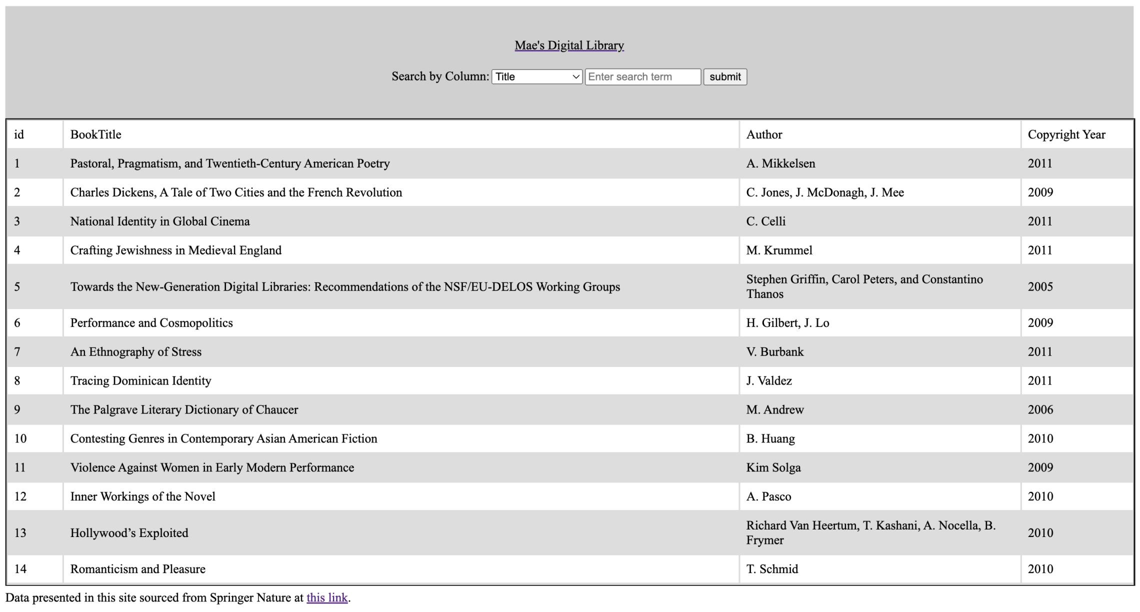


Figure 3.2: Drop-down menu for search column selection.

```
if request.method == "POST":

    # requesting information from the form and storing
        them as variables
    column = request.form["searchColumn"]
    searchTerm = request.form['searchTerm']

    # incorporating the search column and search term into
        the query with string concatanation
    query = "SELECT id, ElectronicISBN, BookTitle, Author,
        CopyrightYear FROM books WHERE " + column + " LIKE
        " + searchTerm + ";"

    # executing the query
    mycursor = mydb.cursor()
    mycursor.execute(query)
    dbhtml = mycursor.fetchall()

    # returning the results to the search page
    return render_template("search.html", dbhtml = dbhtml)
```

Listing 3.5: *POST* method app route for **home.html**.

The *POST* method requests information from the form variables and creates a SQL query which then passes this information to **search.html**. The resulting query results are then displayed as seen in Figure 3.3.



Mae's Digital Library

Search by Column: Title Enter search term submit

id	BookTitle	Author	Copyright Year
1	Pastoral, Pragmatism, and Twentieth-Century American Poetry	A. Mikkelsen	2011
2	Charles Dickens, A Tale of Two Cities and the French Revolution	C. Jones, J. McDonagh, J. Mee	2009
3	National Identity in Global Cinema	C. Celli	2011
4	Crafting Jewishness in Medieval England	M. Krummel	2011
5	Towards the New-Generation Digital Libraries: Recommendations of the NSF/EU-DELOS Working Groups	Stephen Griffin, Carol Peters, and Constantino Thanos	2005
6	Performance and Cosmopolitics	H. Gilbert, J. Lo	2009
7	An Ethnography of Stress	V. Burbank	2011
8	Tracing Dominican Identity	J. Valdez	2011
9	The Palgrave Literary Dictionary of Chaucer	M. Andrew	2006
10	Contesting Genres in Contemporary Asian American Fiction	B. Huang	2010
11	Violence Against Women in Early Modern Performance	Kim Solga	2009
12	Inner Workings of the Novel	A. Pasco	2010
13	Hollywood's Exploited	Richard Van Heertum, T. Kashani, A. Nocella, B. Frymer	2010
14	Romanticism and Pleasure	T. Schmid	2010

Data presented in this site sourced from Springer Nature at this link.

Figure 3.3: Search page.

The search page uses the *POST* method to perform searches similarly to the one described in **home.html** in Listing 3.5. The html page retrieves the query results as `dbhtml`, provided by the return statement in the app route, and displays the search results in a table with the code in Listing 3.6. The code on line 8, `{% for row in dbhtml %}`, allows us to reference the information returned from the query passed from **app.py** on a record by record basis.

```
1  <table id="fullTable" border="1" style="border-spacing: 0;
      width:100%;">
2
3      <!-- establishing widths for each column -->
4      <tr align="center">
5          <td width="5%">id</td>
6          <td width="60%">BookTitle</td>
7          <td width="25%">Author</td>
8          <td width="10%">Copyright Year</td>
9      </tr>
10
11     <!-- inputting information from the SQL query -->
12     {% for row in dbhtml %}
13         <tr>
14             <td id="{{row.0}}" onclick='details({{row.0}})
                   '>{{row.0}}</td>
```

```
15              <td id="{{row.0}}" onclick='details({{row.0}})
                    '>{{row.2}}</td>
16              <td id="{{row.3}}">{{row.3}}</td>
17              <td>{{row.4}}</td>
18          </tr>
19      {% endfor %}
20  </table>
```

Listing 3.6: Displaying SQL query results in **search.html**.

The elements in the body of the table, in lines 14 through 17, reference the rows in the resulting table via `{{row.n}}`. We can use this method of retrieving information from the table to create a link specific to each book's details page. The `id` tag in rows 14 and 15 stores each book's id number to be used when clicking on the title. The id tag on row 16 creates a link to display all books by that author in a search.

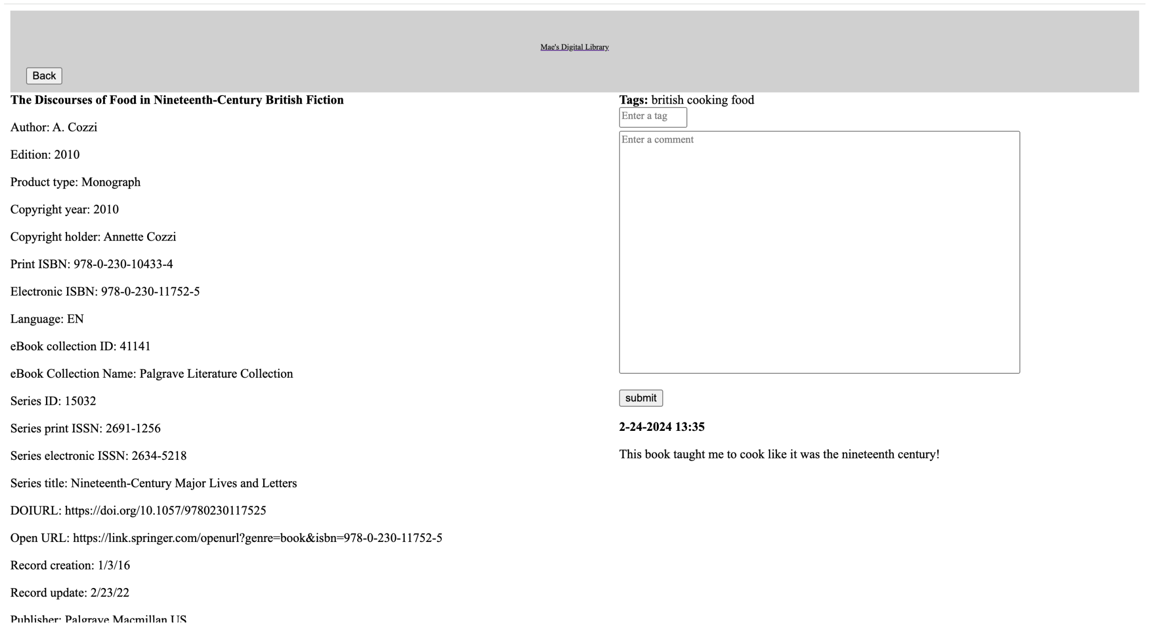
The details page shows the user all bibliographical information present in the database relating to the book. This is obtained by retrieving the book's *id* from the URL, as seen in line 4 of Listing 3.7.

```
1   @app.route('/details.html', methods=['GET','POST'])
2   def display():
3       if request.method == "GET":
4
5           # retrieving book id from the URL and creating a
                query
6           id = request.args.get("q")
7           bookInfoQuery = "SELECT␣*␣FROM␣books␣WHERE␣id␣=␣'"
                + id + "';"
8           mycursor = mydb.cursor()
9           mycursor.execute(bookInfoQuery)
10          bookdb = mycursor.fetchall()
11
12          # retrieving preexisting tags and comments for the
                book
13          commentQuery = "SELECT␣*␣FROM␣comments␣WHERE␣
                book_id␣=␣'" + id + "';"
14          mycursor = mydb.cursor()
15          mycursor.execute(commentQuery)
16          commentdb = mycursor.fetchall()
17
```

```
18        tagQuery = "SELECT Tag FROM tags WHERE book_id = '
              " + id + "';"
19        mycursor = mydb.cursor()
20        mycursor.execute(tagQuery)
21        tagdb = mycursor.fetchall()
22
23        return render_template("details.html", bookdb =
              bookdb, commentdb = commentdb, tagdb = tagdb,
              id = id)
```

Listing 3.7: Retrieving URL arguments in **app.py**.

The `return` statement on line 23 passes the resulting information from the query to the page for display, similarly to what we implemented in Listing 3.6. The page displays bibliographical information as well as user-generated tags and comments, as seen in Figure 3.4.



The Discourses of Food in Nineteenth-Century British Fiction

Author: A. Cozzi

Edition: 2010

Product type: Monograph

Copyright year: 2010

Copyright holder: Annette Cozzi

Print ISBN: 978-0-230-10433-4

Electronic ISBN: 978-0-230-11752-5

Language: EN

eBook collection ID: 41141

eBook Collection Name: Palgrave Literature Collection

Series ID: 15032

Series print ISSN: 2691-1256

Series electronic ISSN: 2634-5218

Series title: Nineteenth-Century Major Lives and Letters

DOIURL: https://doi.org/10.1057/9780230117525

Open URL: https://link.springer.com/openurl?genre=book&isbn=978-0-230-11752-5

Record creation: 1/3/16

Record update: 2/23/22

Publisher: Palgrave Macmillan US

Tags: british cooking food

2-24-2024 13:35

This book taught me to cook like it was the nineteenth century!

Figure 3.4: Book details page.

The content on the left side of the page is originally sourced from Springer Nature [1]. The structure of the data is available in Appendix A. The content on the right side of the page comes from user-generated comments and tags, stored in separate tables in the database. On this page, users can enter tags and comments, go back the the application home page, or return to the search they just made. The SQL connection will be accessed to display a few fields in the search results as well as all bibliographical information.

The code in Listing 3.8 creates the relational table `books`. Lines 2 through 28 create column names for our variables of interest from the dataset supplied by Springer Nature. These lines give SQL information on the column name, the type of variable stored, and the status of null values for that variable. For example, line 2 creates a column entitled ‘BookTitle,’ which stores strings no larger than 250 characters and does not accept null values.

```
1  CREATE TABLE ‘books‘ (
2    ‘BookTitle‘ varchar(250) NOT NULL,
3    ‘Author‘ varchar(250) NOT NULL,
4    ‘Edition‘ varchar(20) DEFAULT NULL,
5    ‘ProductType‘ varchar(250) DEFAULT NULL,
6    ‘CopyrightYear‘ varchar(4) NOT NULL,
7    ‘CopyrightHolder‘ varchar(250) NOT NULL,
8    ‘PrintISBN‘ varchar(20) DEFAULT NULL,
9    ‘ElectronicISBN‘ varchar(20) NOT NULL,
10   ‘Language‘ varchar(250) NOT NULL,
11   ‘eBookCollectionID‘ varchar(20) DEFAULT NULL,
12   ‘eBookCollectionEnglishName‘ varchar(250) DEFAULT NULL,
13   ‘SeriesID‘ varchar(10) DEFAULT NULL,
14   ‘SeriesPrintISSN‘ varchar(20) DEFAULT NULL,
15   ‘SeriesElectronicISSN‘ varchar(20) DEFAULT NULL,
16   ‘SeriesTitle‘ varchar(250) DEFAULT NULL,
17   ‘VolumeNumber‘ varchar(10) DEFAULT NULL,
18   ‘DOIURL‘ varchar(250) NOT NULL,
19   ‘OpenURL‘ varchar(250) DEFAULT NULL,
20   ‘Marc21RecordDate‘ varchar(250) NOT NULL,
21   ‘Marc21ChangeDate‘ varchar(250) DEFAULT NULL,
22   ‘Publisher‘ varchar(250) NOT NULL,
23   ‘Imprint‘ varchar(250) NOT NULL,
24   ‘OpenAccess‘ varchar(250) DEFAULT NULL,
25   ‘ReferenceWorkType‘ varchar(250) DEFAULT NULL,
26   ‘tags‘ varchar(250) DEFAULT NULL,
27   ‘id‘ int(11) NOT NULL
28 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
29
30
```

```
31 INSERT INTO ‘books‘ (‘BookTitle‘, ‘Author‘, ‘Edition‘, ‘
     ProductType‘, ‘CopyrightYear‘, ‘CopyrightHolder‘, ‘
     PrintISBN‘, ‘ElectronicISBN‘, ‘Language‘, ‘
     eBookCollectionID‘, ‘eBookCollectionEnglishName‘, ‘
     SeriesID‘, ‘SeriesPrintISSN‘, ‘SeriesElectronicISSN‘, ‘
     SeriesTitle‘, ‘VolumeNumber‘, ‘DOIURL‘, ‘OpenURL‘, ‘
     Marc21RecordDate‘, ‘Marc21ChangeDate‘, ‘Publisher‘, ‘
     Imprint‘, ‘OpenAccess‘, ‘ReferenceWorkType‘, ‘tags‘, ‘
     id‘) VALUES (. . .)
```

Listing 3.8: Creating the table **books** in MySQL.

Data was transferred into the books table from a Comma Separated Values (CSV) file that contained the data from Springer Nature [1].

Some of the entries in the metadata contain null fields. For this application, we will need to ensure that a few variables do not contain null values across all entries. This is done by using the argument `NOT NULL`. Thus, when inserting data into this table, if the fields are null the entry will be skipped. This means that the resulting table will contain entries with values for every variable specified `NOT NULL`. This does mean that a small amount of data will be lost in the process of importing from the CSV file.

As described in section 2.8.3, SQL allows for the manipulation of relational databases. This application will use relational databases to store both metadata and user-supplied comments or tags. The metadata and user-supplied information are displayed on the site using a connection to a MySQL database, created with the SQL statement in **app.py**, shown in Listing 3.10 The data is imported into SQL in line 31 of Listing 3.8.

SQL allows us to store information in multiple tables. Figure 3.5 shows the tables used in the schema. There are three tables used to store information for the application: **books**, **comments**, and **tags**. These tables are related to one another by the column *id*. In **books**, *id* is the primary key, which increments by one for each entry. This ensures that each book's *id* is unique. The book's *id* is referenced in

comments and **tags** by *book_id*, which is the foreign key. This allows us to display all relevant metadata, tags, and comments on the details page for each book.

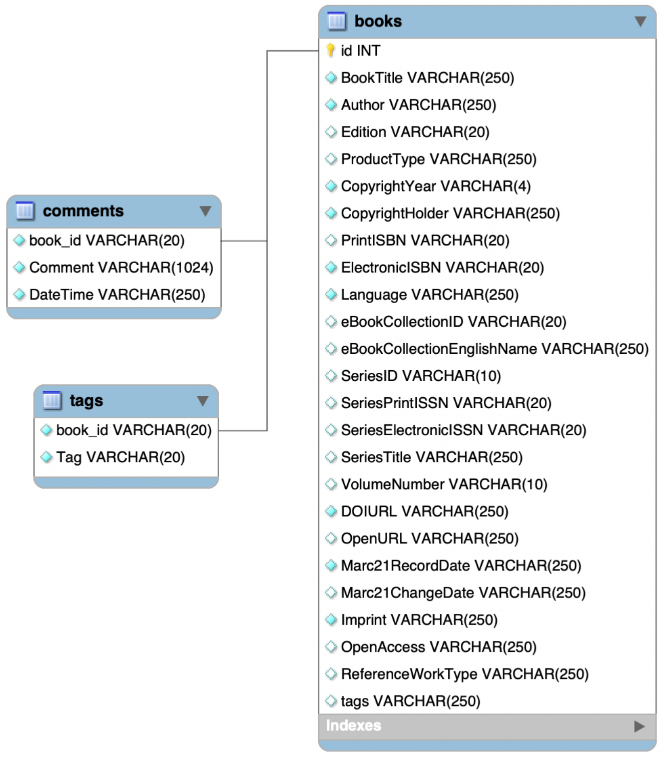


Figure 3.5: Tables used in SQL.

Figure 3.5 displays the tables present in the SQL database. The **books** table, created with the SQL query in Listing 3.8, is used in the **details.html** and in part in **search.html**. Listing 3.9 includes the mySQL query to create two sub-tables: **comments** and **tags**.

```
CREATE TABLE comments (
    book_id varchar(20) NOT NULL,
    Comment varchar(1024) NOT NULL,
    DateTime varchar(250) NOT NULL,
    User varchar(20) NOT NULL
);

CREATE TABLE tags (
    book_id varchar(20) NOT NULL,
    Tag varchar(20) NOT NULL
);
```

Listing 3.9: Creating the tables **tags** and **comments** in MySQL.

To connect with the database, we first need to use MySQL connectors before making queries in Python. Listing 3.10 shows the code used to make the connections in Python.

```
mydb = mysql.connector.connect(
    host = "localhost",
    user = "DBUSERNAME",
    password = "DBPASSWORD",
    database = "bookLib",
    auth_plugin='mysql_native_password'
)
```

Listing 3.10: Connecting to MySQL in **app.py**.

This connection to the SQL database allows us to make `SELECT` and `INSERT` queries to our tables within **app.py**. Queries to SQL are made primarily through **app.py**. Listing 3.11 displays the queries used in app routes **search.html** and **details.html**. These queries are stored as strings and passed to the `execute()` function, as shown in Listing 3.12. Note that some queries include placeholders for information obtained from the *POST* request forms or URL information.

```
SELECT ElectronicISBN, BookTitle, Author FROM books;
SELECT ElectronicISBN, BookTitle, Author FROM books WHERE
    BookTitle like '%[searchTerm]%';
SELECT ElectronicISBN, BookTitle, Author FROM books WHERE
    ElectronicISBN like '%[searchTerm]%';
SELECT ElectronicISBN, BookTitle, Author FROM books WHERE
    Author like '%[searchTerm]%';
SELECT ElectronicISBN, BookTitle, Author FROM books WHERE
    BookTitle like'%[searchTerm]%' or WHERE ElectronicISBN
    like'%[searchTerm]%' or WHERE Author like '%[searchTerm
    ]%';
SELECT * FROM books WHERE ElectronicISBN = '[isbn]';
SELECT * FROM comments WHERE ISBN = '[isbn]';
```

Listing 3.11: SQL queries used in **app.py**.

```
query = "SELECT * FROM books;"
mycursor = mydb.cursor()
mycursor.execute(query)
dbhtml = mycursor.fetchall()
return render_template("details.html", dbhtml = dbhrml)
```

Listing 3.12: Running queries in **app.py** in app routes.

Listing 3.12 shows the method used to make queries to the SQL database. The functions used come from the Python library `sql.connector`. We then render the template, passing the query results as `dbhtml`.

3.2 User Interface

In the creation of this application, we must keep in mind the elements of user interface (UI) design to effectively render an application which clearly communicates to its users its usability.

3.2.1 Aesthetics

In the Cascading Style Sheets (CSS) site styling, we established colors and fonts to use in the application to ensure consistency of styling throughout the site. We chose colors for this application that would not distract from the purpose of the site, while also making it look aesthetically pleasing.

3.2.2 Consistency

When using multiple HTML templates, establishing consistency throughout the site can be done with headers, in addition to the CSS site styling. We included the same header tag on all pages to ensure consistency.

3.2.3 Predictability

When creating navigational and other functional aspects to the site, it is important to consider the expectations a user may have before encountering the site. Thus, we need to consider conventions commonly used in user interfaces for user interactions. Examples of these used in this site include the use of a dropdown menu for searching and the submit button for adding comments.

3.2.4 Recovery

When adding comments and tags to the site, it is important to allow the user the opportunity to review their comment before submitting it. This allows users to read through the comment or tag in advance of submitting it to the database in case they change their mind.

3.3 Scrolling

In this application, we will be using scrolling to allow users to see all results in the table. This ensures that all results can be seen.

3.4 Usability Testing

To get an idea of how the intended audience will interact with the software, usability testing is necessary. Through the usability testing process, we can watch users interact with the application and identify where their expectations of the site differ from the affordances. The purpose of the study is to identify pertinent disjunctions between expectation and functionality when interacting with the site.

During testing, the user will be given a list of tasks to perform on the site with a questionnaire to follow. The user is asked to perform the tasks to the best of their ability, taking note of places in which the software did not work as expected. The research protocol for usability testing with a full list of tasks is included in Appendix B, along with a copy of the proposal and the Human Subjects Research Committee (HSRC) approval obtained for the study.

The intended audience of the software includes students, faculty, and librarians in a university environment. Thus, we will be including individuals who fit these identities in our study. For this study, five participants were chosen from the target population at the College of Wooster.

When considering tasks to include for the usability study, we should consider all affordances of the site so we can maximize the information learned about the usability of the site. This site allows users to search, comment, tag, and otherwise navigate the site, so tasks must encompass all functionality. For this usability study, tasks asked of participants include the following:

1. Find a book published in the year 2015.
2. Go to that book's page.
3. Add a comment to the book's page.
4. Go back to the home page.
5. Add a tag to a book written by G. Atkins.

Upon completion of the tasks, participants are asked to answer questions based on their experience navigating the site. Most questions utilize a Likert scale, with one and five being *strongly disagree* and *strongly agree*, respectively. These questions are as follows:

1. I had no trouble understanding what this application's purpose is.
2. The buttons in this application clearly inform what is being clicked on.
3. Navigating the application worked as expected.
4. The colors used on this site were aesthetically pleasing and did not distract from the purpose.
5. Interactive features on this site (adding comments and tags) worked as expected.
6. The search features worked as expected.
7. Please rate this product based on your overall experience (1 = bad, 5 = excellent)
8. Did you find it difficult to complete any task? If so, please elaborate.

Based on the feedback provided in usability testing, we can then make changes to the software to better serve the user where possible.

CHAPTER 4

Results

To test the overall user experience of the site, a usability study is necessary to gain an understanding of how the targeted user group will interact with the software. This will allow us to make changes to the application to better fit the needs of the audience. The tasks included in the usability testing were selected to maximize interaction with the software. Tasks asked of the participants are as follows:

1. Find a book published in the year 2015.
2. Go to that book's page.
3. Add a comment to the book's page.
4. Go back to the home page.
5. Add a tag to a book written by G. Atkins.

Upon completion of the tasks, the participant was asked to answer a number of questions regarding their experience using the site. Table 4.1 shows the recorded responses to the questions posed in the usability testing. The first several questions are on a Likert scale, with one being *strongly disagree* and five being *strongly agree*. The last question is open-ended and allows the participant to provide comments or other information not encompassed by the Likert scale questions. Questions asked in the usability study are as follows:

1. I had no trouble understanding what this application's purpose is.

2. The buttons in this application clearly inform what is being clicked on.
3. Navigating the application worked as expected.
4. The colors used on this site were aesthetically pleasing and did not distract from the purpose.
5. Interactive features on this site (adding comments and tags) worked as expected.
6. The search features worked as expected.
7. Please rate this product based on your overall experience (1 = bad, 5 = excellent)
8. Did you find it difficult to complete any task? If so, please elaborate.

These questions aim to ensure the site is as user-friendly as possible, while also following the principles of UI design. Table 4.1 shows the data collected from questions 1-7. Survey responses in full are included in Appendix B.

Question Number	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Average Response
1	0	0	1	1	3	4.4
2	0	0	0	0	5	5.0
3	0	0	1	2	2	4.2
4	0	0	3	1	1	3.6
5	0	0	0	4	1	4.2
6	0	1	0	0	4	4.4
7	0	0	1	1	3	4.4

Table 4.1: Usability study survey results.

Question 1 stated “I had no trouble understanding what this application’s purpose is." Three participants responded *strongly agree*, and the other two participants responded *agree* and *neutral*. This indicates that the purpose of the application could have been more clearly indicated through documentation on the home page of the site.

Question 2 stated “The buttons in this application clearly inform what is being clicked on." All survey participants responded to this question with *strongly agree*.

Question 3 stated “Navigating the application worked as expected." Two participants responded with *strongly agree*, two responded with *agree*, and one participant responded with *neutral*. This indicates that the usability of this software can be improved by adding more signifiers to communicate navigational aspects within the application.

Question 4 stated “The colors used in this site were aesthetically pleasing and did not distract from the purpose." Three users responded to this question with *neutral*, while the other two participants responded with *agree* and *strongly agree*. Based on these responses, we can conclude that the site could benefit a more interesting color palette to make it interesting. A few participants suggested the use of color on the site.

Question 5 stated “Interactive features on the site (adding comments and tags) worked as expected." Four participants responded with *agree*, and the other with *strongly agree*.

Question 6 stated “The search features worked as expected." Four participants responded *strongly agree*, while the other participant responded *disagree*. This outlier would be cause for concern, however, due to the extended use of Flask the SQL connection failed for a portion of this participant’s usability study, suggesting that this data point was unreliable for the purposes of this study.

Question 7 stated “Please rate this product based on your overall experience (1 = bad, 5 = excellent)." The average response for this question was 4.4, indicating that most participants had a positive experience with the site.

In addition to the Likert scale questions, we asked the user “Did you find it difficult to complete any task? If so, elaborate." This open-ended question provided a space for participants to provide comments on their experience with the software. One participant commented on the signifiers used in the software, “I found it difficult to tell when I had the cursor over a hyperlink."

In response to question 8, a couple of participants provided positive feedback to the site relating to elements of the software they liked. One participant said it was “similar to Wooster's access services software." Another participant wrote about the aesthetics of the home page, a “clear, minimal display ... [that is] not too distracting and [is] straightforward."

Based on the responses to the usability survey, we can make conclusions about what changes to make to the application to enhance user experience. These changes include implementing cursor changes when hovering on a link and allowing the user to enter tags and comments using the $\boxed{\text{enter}}$ key. Another possible change would be to include more information about the purpose of this project on the home page of the site. We can also implement more aesthetic-related changes to make the site to boost user engagement.

CHAPTER 5

Conclusion

This project aimed to create an Integrated Library System (ILS) that focuses on user collaboration in an academic setting. The software allows users to search, tag, and provide academic discussion of academic literature in a database.

5.1 Problems Encountered

In the process of creating the software for this application, issues with converting the data from the CSV file into the SQL database, which caused us to use a smaller database than planned.

The lack of time given to work on the project contributed to its limited features. If given more time to complete the software, it would have featured a wider array of functionality, including login for multiple user types, creating a page where users can see all the comments they have posted, and allowing administrators to edit the database.

5.2 Future Work

Future work on this software includes added functionality for managing or otherwise updating the books in the library, if the book is physically stored and not a digital copy. This would be implemented with an additional table in the schema which would hold all relevant information regarding the book's physical location. This would allow administrators to identify whether a book is out on loan, to whom, and the date it is expected to be returned to the library.

To ensure library records are changed only by those authorized to do so, it is imperative to include functionality for multiple logins. In this case, we would use an additional table to store user login information such as *username*, *password*, *email*, and *user_type*. Adding functionality for users to log into the site allows us to manage who has the authority to update records in the system. This would also allow us to attribute comments to the user who wrote the comment on the book details page.

Libraries often gain access to new material to be included in their database. Thus, we can incorporate functionality for creating, updating, and deleting entries in the database for administrators and librarians.

Another managerial aspect to include in future iterations is authority control for tagging and commenting. The incorporation of authority control would reduce redundancy and repetition in tags as well as prevent users from posting inappropriate content to the site or other content that violates community guidelines.

To expand the effectiveness of the search functionality in this software, it can be beneficial to use stemming in the search process. The use of stemming would allow for more results when performing a search on the site. Stemming is a process in which words are broken down to their root, so this would increase the precision of meaning rather than phrasing when searching for books in a database.

With the current version of the application, a search for a title that does not appear in the database yields an empty table. To better enhance user experience, we can let the user know that there were no matches in the database and suggest a different search based on the search entered in case of incorrect spelling. This can be done using natural language processing.

While there are many further implementations to be made in this software, it provides a platform for academic discussion in a Library Management System. The site allows students to collaboratively contribute academic discussion and tags to books in a library.

APPENDIX A

Database

The dataset presented in this software is sourced from Springer Nature, a member of the Open Access Scholarly Publishers Association. This dataset includes metadata relating to books discussing literature, cultural, and media studies. This metadata includes the books' titles, authors, languages, and ISBNs, as well as other relevant bibliographical information, described in Table A.1 [1].

Column Name	Type	Description
BookTitle	char	Title of book
Author	char	Author of book
Edition	int	Year the book was published
ProductType	char	Type of media
CopyrightYear	int	Year copyrighted
CopyrightHolder	char	Name of copyright holder
PrintISBN	int	ISBN associated with print version of book
ElectronicISBN	int	ISBN associated with electronic version of book
Language	char	Language in which the book is displayed
eBookCollectionID	int	Identification number of the collection to which the book belongs
eBookCollectionEnglishName	char	Name of the collection to which the book belongs
SeriesID	int	Identification number of the series to which the book belongs

Column Name	Type	Description
SeriesPrintISSN	int	ISSN attributed to the print version of the series to which the book belongs
SeriesElectronicISSN	int	ISSN attributed to the electronic version of the series to which the book belongs
SeriesTitle	char	Name of the series to which the book belongs, if any
VolumeNumber	int	Volume number of book
DOIURL	char	Persistent URL pointing to the text on SpringerNature
OpenURL	char	URL pointing to the text on SpringerNature
Marc21RecordDate	int	Date at which the MARC record for the book was created
Marc21ChangeDate	int	Date at which the MARC record for the book was last changed
Publisher	char	Name of company that published the book
Imprint	char	Company the book was printed under
OpenAccess	boolean	Whether or not the book is open access
ReferenceWorkType	char	Type of work

Table A.1: Description of metadata displayed in the site, sourced from SpringerNature [1].

It is important to note the difference between *DOIURL* and *OpenURL*. *DOIs* are unbreakable, persistent links that is associated with an *OpenURL*. *OpenURLs* are subject to change in the case that a publisher may change names or otherwise change. *DOIs* point to the URL at which the content is stored [71], and are updated to point to the correct URL upon change.

APPENDIX B

Human Research and Usability Testing

B.1 Consent to participate in a research study at The College of Wooster

An Integrated Library System that Focuses on Collaboration: Bringing Books and Computers Together

Principal Investigator: Mae Koger, Department of Mathematical & Computational Sciences

Purpose

You are being asked to participate in a usability study. I am investigating the usability of an Integrated Library System for the purpose of collaboration in an academic setting.

Procedures

If you decide to volunteer, you will be asked to execute tasks on the application and answer several questions about the usability of the site. Each interview will take approximately 20-45 minutes to complete.

Risks

There are no associated risks with participating in this study.

Benefits

There are no direct benefits to you for your participation. An indirect benefit is that I will learn more about how this software is interacted with and perceived by users.

Compensation

Participants will not receive any compensation for participating in this study.

Confidentiality

Any information you give will be held confidential. Unique name/number codes will be erased. This file will be destroyed once all data is collected. Thus, all data will become anonymous at the conclusion of the study.

Costs

There is no cost to you beyond the time and effort required to complete the procedure described above.

Right to Refuse or Withdraw

You may refuse to participate in the study. If you decide to participate, you may change your mind about being in the study and withdraw at any point during the experiment.

Questions

If you have any questions, please ask me. If you have additional questions later, you can contact me by email at mkoger24@wooster.edu You may also contact my advisor, Thomas Montelione, at tmontelione@wooster.edu.

Consent

Your signature below will indicate that you have decided to volunteer as a research subject, that you have read and understand the information provided above, and that you are at least 18 years of age. Signature of participant: ______________________

Date: ______________

You will be provided a copy of this form.

B.2 Research Protocol

Background information

The purpose of this project is to focus on the use of digital libraries as software to facilitate learning and easy retrieval of materials in an academic context as well as collaboration within this environment. In the creation of this digital library, it is crucial to understand how digital libraries should function in an educational environment as well as explore the features digital libraries currently employ to facilitate learning. This software will allow all users—whether student, librarian, or professor—to effectively communicate with one another.

Specific aims of your research

This study aims to determine the best use of the principles of user interface (UI) design through provided tasks to complete on the application, followed by questions detailing ease of use.

Location where the research will be conducted

Interviews will be conducted at The College of Wooster campus or an environment of the participants' choice. The interviews will be conducted in person or virtually using Microsoft Teams.

With whom the data and/or conclusions will be shared

I will share my data with my advisor Thomas Montelione as needed, but otherwise data will be confidential. Conclusions will be presented my Independent Study with summary information available on Open Works.

Methodology of your study

Participants will be given a list of tasks to complete on the application. Upon completion of the tasks, participants will be asked survey questions detailing their experience using the application. All interviews are voluntary and anonymous.

If you are recording data via audio or video recording devices, then you must include a description of how/where these materials will be stored, coded, encrypted, and/or destroyed upon conclusion of the study.

Data will be stored on a password protected computer and destroyed upon completion of the software. Any consent forms will be stored on a password protected computer and stored separately from the data.

B.3 Interview Questions

Tasks

1. Find a book published in the year 2015.
2. Go to that book's page.
3. Add a comment to the book's page.
4. Go back to the home page.
5. Add a tag to a book written by G. Atkins.

Survey

Circle the choice that best describes yourself.

Scale: 1 – Strongly Disagree 2 – Disagree 3 – Neutral 4 – Agree 5 – Strongly Agree

1. I had no trouble understanding what this application's purpose is.

   1 2 3 4 5

2. The buttons in this application clearly inform what is being clicked on.

   1 2 3 4 5

3. Navigating the application worked as expected.

   1 2 3 4 5

4. The colors used on this site were aesthetically pleasing and did not distract from the purpose.

   1 2 3 4 5

5. Interactive features on the site (adding comments and tags) worked as expected.

   1 2 3 4 5

6. The search features worked as expected.

   1 2 3 4 5

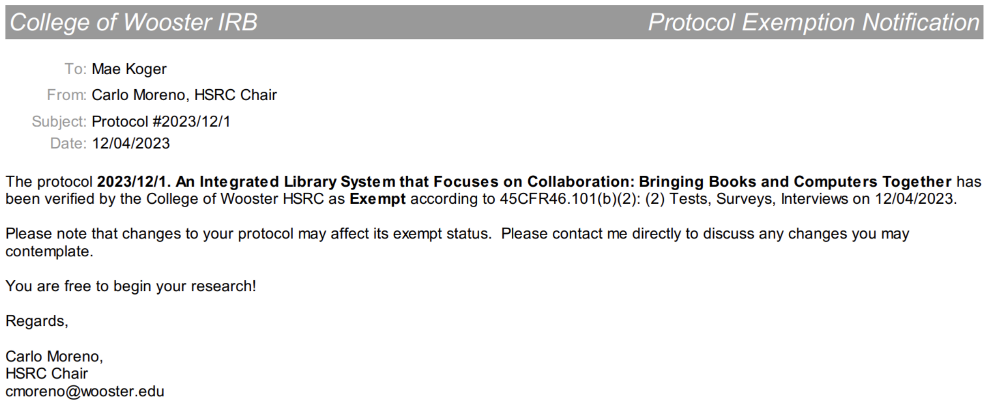
7. Please rate this product based on your overall experience (1 = bad, 5 = excellent)

   1 2 3 4 5

8. Did you find it difficult to complete any task? If so, please elaborate.

B.4 Human Subjects Research Committee Approval

The usability study conducted for this project received approval from the Human Subjects Research Committee before conducting testing. A copy of the approval notification is presented in Figure B.1.



College of Wooster IRB Protocol Exemption Notification

To: Mae Koger
From: Carlo Moreno, HSRC Chair
Subject: Protocol #2023/12/1
Date: 12/04/2023

The protocol **2023/12/1. An Integrated Library System that Focuses on Collaboration: Bringing Books and Computers Together** has been verified by the College of Wooster HSRC as **Exempt** according to 45CFR46.101(b)(2): (2) Tests, Surveys, Interviews on 12/04/2023.

Please note that changes to your protocol may affect its exempt status. Please contact me directly to discuss any changes you may contemplate.

You are free to begin your research!

Regards,

Carlo Moreno,
HSRC Chair
cmoreno@wooster.edu

Figure B.1: HSRC approval for conducting the usability study.

References

[1] 2016. URL: https://metadata.springernature.com/metadata/books (cit. on pp. 2, 28, 35, 37, 51–52).

[2] 2023. URL: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html (cit. on p. 18).

[3] 2023. URL: http://pnwx.com/ (cit. on p. 21).

[4] 2023. URL: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C36&q=what+is+an+integrated+library+system&oq=what+is+an+int (cit. on p. 25).

[5] 2023. URL: https://wooster.presence.io/ (cit. on p. 25).

[6] 2023. URL: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html.

[7] 2023. URL: https://libguides.eur.nl/informationskillssearchmethods/google#:~:text=Internet%20search%20engines%20like%20Google,found%20using%20internet%20search%20engines..

[8] Daniel J Abadi. "Data management in the cloud: Limitations and opportunities." In: *IEEE Data Eng. Bull.* 32.1 (2009), pp. 3–12 (cit. on pp. 15–16).

[9] Tobi Adebesin and Olaiya Folorunsho. "Library Management System". PhD thesis. Federal University Oye Ekiti, 2015 (cit. on pp. 1, 7).

[10] Sana Afreen. *What Is Cloud Computing Architecture: Benefits, Components More*. June 2020. URL: `https://www.simplilearn.com/tutorials/cloud-computing-tutorial/cloud-computing-architecture` (cit. on p. 15).

[11] Monjur Ahmed and Mohammad Ashraf Hossain. "Cloud computing and security issues in the cloud". In: *International Journal of Network Security & Its Applications* 6.1 (2014), p. 25 (cit. on pp. 15–16).

[12] Leif Andresen. "After MARC–what then?" In: *Library hi tech* 22.1 (2004), pp. 40–51 (cit. on pp. 5–6).

[13] Henriette D Avram. "The MARC Pilot Project. Final Report." In: (1968) (cit. on p. 5).

[14] David Bainbridge et al. "Dynamic digital library construction and configuration". In: Springer. 2004, pp. 1–13 (cit. on pp. 8–9).

[15] Janet L Balas. "Developing Library Collections for a Wired World." In: *Computers in libraries* 20.6 (2000), pp. 61–63 (cit. on p. 9).

[16] Syeda Hina Batool. "Status of technological competencies: a case study of university librarians". In: *Library Philosophy and Practice* (2010), p. 1 (cit. on p. 6).

[17] Michael Bernard et al. "Paging vs. scrolling: Examining ways to present search results". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 46. 14. SAGE Publications Sage CA: Los Angeles, CA. 2002, pp. 1296–1299 (cit. on p. 25).

[18] N Uday Bhaskar et al. "General principles of user interface design and websites". In: *International Journal of Software Engineering (IJSE)* 2.3 (2011), pp. 45–60 (cit. on pp. 21–23).

[19] Vannevar Bush et al. "As we may think". In: *The Atlantic Monthly* 176.1 (1945), pp. 101–108 (cit. on p. 7).

[20] Leonardo Candela, Donatella Castelli, and Pasquale Pagano. “History, evolution and impact of Digital Libraries”. In: *ResearchGate* (2011) (cit. on pp. 7, 9–10).

[21] Matthew Carey, Daniel Heesch, and Stefan Rüger. “Info navigator: A visualization tool for document searching and browsing”. In: (2003) (cit. on p. 12).

[22] Tanaka Chingonzo. “Imali: A Technical Analysis Mobile Application in React Native”. In: (2021) (cit. on p. 17).

[23] Andrew Dillon. “User interface design”. In: London: Macmillan, 2003 (cit. on p. 19).

[24] Tharam Dillon, Chen Wu, and Elizabeth Chang. “Cloud computing: issues and challenges”. In: *2010 24th IEEE international conference on advanced information networking and applications*. Ieee. 2010, pp. 27–33 (cit. on p. 15).

[25] Ben Foltz. “Built To Scale: An Exploration Into Full Stack Development Strategies Through The Construction Of A Web Application”. In: (2022) (cit. on p. 17).

[26] Stephen Griffin, Carol Peters, and Costantino Thanos. “Towards the new-generation digital libraries: recommendations of the NSF/EU-DELOS working groups”. In: *International Journal on Digital Libraries* 5.4 (2005), p. 253 (cit. on p. 4).

[27] Robert L Grossman. “The case for cloud computing”. In: *IT professional* 11.2 (2009), pp. 23–27 (cit. on p. 16).

[28] Cornelia Gyorödi, Robert Gyorödi, and Roxana Sotoc. “A comparative study of relational and non-relational database models in a Web-based application”. In: *International Journal of Advanced Computer Science and Applications* 6.11 (2015), pp. 78–83.

[29] Joacim Hansson. “Just collaboration or really something else? On Joint-use libraries and normative institutional change with two examples from Sweden”. In: *Library trends* 54.4 (2006), pp. 549–568.

[30] Jan L Harrington. *Relational database design and implementation*. Morgan Kaufmann, 2016 (cit. on p. 17).

[31] Michael James Hernandez. *Database design for mere mortals: a hands-on guide to relational database design*. Pearson Education, 2013 (cit. on p. 18).

[32] Arnold Hirshon. “Library strategic alliances and the digital library in the 1990s: The OhioLINK experience”. In: *The Journal of academic librarianship* 21.5 (1995), pp. 383–386 (cit. on p. 4).

[33] Mu-hsuan Huang and Yu-Wei Chang. “A comparative study of interdisciplinary changes between information science and library science”. In: *Scientometrics* 91.3 (2012), pp. 789–803.

[34] Tim Huynh, Hiep Luong, and Kiem Hoang. “Integrating bibliographical data of computer science publications from online digital libraries”. In: *Intelligent Information and Database Systems: 4th Asian Conference*. 2012, pp. 226–235.

[35] D Curtis Jamison. “Structured query language (SQL) fundamentals”. In: *Current protocols in bioinformatics* 1 (2003), pp. 9–2 (cit. on p. 18).

[36] Shyamli Jha. *What is Cloud Computing and Who Uses Cloud Services?* May 2020. URL: https://www.simplilearn.com/tutorials/cloud-computing-tutorial/what-is-cloud-computing (cit. on p. 15).

[37] Jeff Johnson. *Designing with the mind in mind: simple guide to understanding user interface design guidelines*. Morgan Kaufmann, 2020 (cit. on pp. 21–22).

[38] Paul Kahn and Krzysztof Lenk. “Design: principles of typography for user interface design”. In: *interactions* 5.6 (1998), p. 15 (cit. on p. 23).

[39] Diane Kelly and Leif Azzopardi. "How many results per page? A study of SERP size, search behavior and user experience". In: *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 2015, pp. 183–192 (cit. on pp. 24–25).

[40] Indira Koneru. "Integrated library system: Selection and design". In: *DESIDOC Journal of Library & Information Technology* 25.6 (2005).

[41] Mikael Laakso et al. "The development of open access journal publishing from 1993 to 2009". In: *PloS one* 6.6 (2011), e20961 (cit. on p. 14).

[42] Carl Lagoze et al. "Fedora: an architecture for complex objects and their relationships". In: *International Journal on Digital Libraries* 6 (2006), pp. 124–138 (cit. on p. 8).

[43] Carl Lagoze et al. "What is a digital library anymore, anyway?" In: *D-Lib magazine* 11.11 (2005) (cit. on pp. 1, 3, 7, 10).

[44] Margaret G Lyons. "Open access is almost here: navigating through copyright, fair use, and the TEACH Act". In: *The Journal of Continuing Education in Nursing* 41.2 (2010), pp. 57–64 (cit. on p. 13).

[45] Amanda McCormick. "Copyright, fair use and the digital age in academic libraries: A review of the literature". In: *School of Information Student Research Journal* 4.2 (2014), p. 5 (cit. on p. 13).

[46] Dan McCreary and Ann Kelly. "Making sense of NoSQL". In: *Shelter Island: Manning* (2014), pp. 19–20 (cit. on p. 17).

[47] Katharine Morton. "The MARC formats: an overview". In: *The American Archivist* 49.1 (1986), pp. 21–30 (cit. on pp. 5–6).

[48] KR Mulla and M Chandrashekara. "Use of integrated library software: A survey of engineering college libraries in Karnataka". In: *International Journal of Information Science and Management (IJISM)* 8.2 (2010), pp. 99–111.

[49] Laura Bowering Mullen. “Publishers and librarians: New dialogues in challenging times”. In: *Issues in Science & Technology Librarianship* 56 (2009).

[50] AB MySQL. *MySQL*. 2001 (cit. on p. 18).

[51] Don Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013 (cit. on pp. 19, 24).

[52] Jennifer Novia. “Library, archival and museum (LAM) collaboration: Driving forces and recent trends”. In: *Endnotes: The Journal of the New Members Round Table* 3.1 (2012), pp. 1–10 (cit. on p. 10).

[53] Greg Nudelman. *Designing search: UX strategies for ecommerce success*. John Wiley & Sons, 2011 (cit. on p. 26).

[54] Chukwuma Clement Okeji and Okeoghene Mayowa-Adebara. “An evaluation of digital library education in library and information science curriculum in Nigerian Universities”. In: *Digital Library Perspectives* (2020).

[55] Isabella Peters and Wolfgang G Stock. “Folksonomy and information retrieval”. In: *Proceedings of the American Society for Information Science and Technology* 44.1 (2007), pp. 1–28 (cit. on pp. 10–11).

[56] Rebecca R Pressman. “Fair use: Law, ethics and librarians”. In: *Journal of Library Administration* 47.3-4 (2008), pp. 89–110 (cit. on p. 13).

[57] Ling Qian et al. “Cloud computing: An overview”. In: *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings 1*. Springer. 2009, pp. 626–631 (cit. on p. 15).

[58] Randhawa and Sukhwinder. *Open Source Library Management Softwares Item Type Article*. 2023. URL: https://repository.arizona.edu/bitstream/handle/10150/299575/Open+Source+Library+Management+Softwares.pdf?sequence=1 (cit. on pp. 4, 14).

[59] Sukhwinder Randhawa. *Designing the search experience: The information architecture of discovery*. Newnes, 2012.

[60] James Sinclair and Michael Cardew-Hall. "The folksonomy tag cloud: when is it useful?" In: *Journal of Information Science* 34.1 (2008), pp. 15–29 (cit. on pp. 11–12).

[61] Manisha Singh and Gareema Sanaman. "Open source integrated library management systems: comparative analysis of Koha and NewGenLib". In: *The Electronic Library* 30.6 (2012), pp. 809–832 (cit. on pp. 14–15).

[62] Louise F Spiteri. "The structure and form of folksonomy tags: The road to the public library catalog". In: *Information technology and libraries* 26.3 (2007), pp. 13–25.

[63] J Srinivas, K Venkata Subba Reddy, and A Moiz Qyser. "Cloud computing basics". In: *International journal of advanced research in computer and communication engineering* 1.5 (2012), pp. 343–347 (cit. on p. 15).

[64] Thornton Staples, Ross Wayland, and Sandra Payette. "The Fedora Project". In: *D-Lib Magazine* 9.4 (2003), pp. 1082–9873 (cit. on p. 7).

[65] Bryan Stroube. "Literary freedom: Project Gutenberg". In: *XRDS: Crossroads, The ACM Magazine for Students* 10.1 (2003), pp. 3–3 (cit. on pp. 9, 13).

[66] Sandra L Stump and Rosemary L Deegan. "Open Source Opens Doors: Repurposing Library Software to Facilitate Faculty Research and Collaboration". In: *Pennsylvania Libraries: Research & Practice* 1.2 (2013), pp. 162–167 (cit. on p. 15).

[67] Ren Tao. "Research on Cloud Computer and Library Management Challenges". In: Atlantis Press. 2016, pp. 548–551 (cit. on p. 16).

[68] Roy Tennant. "MARC must die". In: *LIBRARY JOURNAL-NEW YORK-* 127.17 (2002), pp. 26–27 (cit. on p. 5).

[69] Martijn Van Welie, Gerrit C Van Der Veer, and Anton Eliëns. "Patterns as tools for user interface design". In: *Tools for Working with Guidelines: Annual Meeting of the Special Interest Group*. Springer. 2001, pp. 313–324 (cit. on p. 20).

[70] Yongming Wang and Trevor A Dawes. "The next generation integrated library system: a promise fulfilled?" In: *Information technology and libraries* 31.3 (2012), pp. 76–84 (cit. on p. 6).

[71] Katie Wilson. "Open URL: linking through the maze of online resources". In: *Online Currents* 19.2 (2004), pp. 3–8 (cit. on p. 52).

[72] Titin Winarti, Jati Kerami, and Sunny Arief. "Determining term on text document clustering using algorithm of enhanced confix stripping stemming". In: *Int. J. Comput. Appl* 157.9 (2017), pp. 8–13 (cit. on p. 25).

[73] Ian H Witten. "The development and usage of the Greenstone digital library software". In: (2008) (cit. on p. 8).

[74] Ian H Witten and David Bainbridge. "Creating digital library collections with Greenstone". In: *Library Hi Tech* 23.4 (2005), pp. 541–560 (cit. on p. 8).

[75] Ian H Witten, David Bainbridge, and David M Nichols. *How to build a digital library*. Morgan Kaufmann, 2009 (cit. on pp. 1, 10, 12).

[76] Ian H Witten et al. "Greenstone: a comprehensive open-source digital library software system". In: *Proceedings of the fifth ACM conference on Digital libraries*. 2000, pp. 113–121 (cit. on p. 8).

[77] Chaojun Yao. "The Application of Computer Science and Technology in Libraries". In: *International Conference on Education, Management, Computer and Society*. Atlantis Press. 2016, pp. 1422–1424 (cit. on p. 9).

[78] Shea-Tinn Yeh and Zhiping Walter. "Critical success factors for integrated library system implementation in academic libraries: A qualitative study". In: *Information Technology and libraries* 35.3 (2016), pp. 27–42.