

TRACKING THE LIBRARY BOOKS USING AN AUTOMATED MANAGEMENT SYSTEM

A. Viswanathan¹, M. Umamaheswari², S. Sasikanth³, M. Sathya⁴,

¹Associate Professor in CSE, KSR College of Engineering, Tiruchengode-637 215

²Assistant Professor in CSE, KSR College of Engineering, Tiruchengode-637 215

³Assistant Professor in ECE, Vivekanandha college of Technology for women

⁴Assistant Professor in CSE K.S.R College of Engineering, Tiruchengode -637215

Email: ¹professorvichu@gmail.com, ²umadeena@gmail.com, ³ssasi44@gmail.com,
⁴sathimanogaran@gmail.com

Abstract:

The Library Management System was developed to monitor the arrival and departure of books from the library as well as student information. While new students are tracked using the Student Entry form, new books are tracked using the Books Entry form. If a student requests a book and it is available, the book is provided to the student. The book's issuance and due date are likewise placed into the Book Issue form under the third option, Book Issue. Depending on how many days the book was not returned to the library, the student may be required to pay a charge. The automation of the library system is the primary goal of this project. The current task is to fully automate the application. To make book issues easier to manage, it is recorded in the database together with stack information. Additionally, it may return books, add new titles, and do availability-based searches for books. The books may be separated into categories to make discovering them simpler. We may also find out how many books are available in the library. The database has the ability to store user data.

1. INTRODUCTION

It is challenging to manually locate and lend books that have been returned due to making or accountability issues. There is a considerable possibility that this record will be lost or misplaced because the borrowed books are documented in a book. This can make it impossible for the library to get the books that were checked out.

Implementing an electronic library management system will provide rapid and efficient administration of library services, as well as simple Utilities literary search and wider access to reading materials by enabling users to download e-books from any location on campus. The automation of the library system is the primary goal of this project. The current task is to fully automate the application. To make book issues easier to manage, it is recorded in the database together with stack information. Additionally, it may return books, add new titles, and do availability-based searches for books. The books may be separated into categories to

make discovering them simpler. We may also find out how many books are available in the library. The database has the ability to store user data.

Information technology (IT) applications of all kinds have changed and facilitated life. In response to the rapid changes in information technology usage, several tools, techniques, and systems have been created and constructed. Information technology, which combines computer and telecommunications technology, enables the creation of new systems and products that help people at home, at work, and in school. In the modern world, several activities may take place at the same time and in the same place, making it necessary to integrate all processes, provide paperless workplaces, and ensure proper management responsibilities.

Due to making or accountability difficulties, it is difficult to manually track and lend books that have been returned. The fact that the books were borrowed and recorded in a book increases the likelihood that this record would be lost or misplaced. This could prevent the library from getting its hands on the borrowed books. Therefore, the goal of this work is to implement an electronic library management system that will make it simple and effective to manage library services, as well as to make it simple to search for literature using utilities, and to increase access to reading materials by enabling users to access e-books from anywhere on campus.

2. RELATED WORKS

Since the system enables library employees to keep track of their patrons and the resources they administer, the majority of the objectives were achieved. All data is now simple to alter due to the nature of electronic storage, making the compilation of reports simple. Finding reading material has become easier since a variety of criteria may be employed to complete the process. The interfaces were user-friendly; thus, retraining was not necessary beyond orientation. Also, presented the ISIS-software roots of the recently published ABCD-software, which has piqued the interest of several informational organizations and libraries in poor nations.

They work to educate them so that they may more accurately assess the significance of the ABCD software for international librarianship and, why not, for their own issues. Many Western librarians, especially the younger ones, have never heard of ISIS, its relevance, or its technological concepts.

The "Free and Open Source" (FOSS) software development movement, which is gaining a lot of support in the library and documentation sectors, will be compared to the (CDS/) ISIS software in this study as a "predecessor." We will emphasize several technical components and traits in this context to show how the program has goals of being "free" and even "open" since its beginnings, which is extremely lengthy in computer science traditions, even though ISIS's full devotion to this movement is recent.

It would be helpful to first discuss systems before talking about library management systems. A system is anything that is made up of components that work together to accomplish a shared goal. To support the administration of library information resources, including their acquisition, representation, and circulation, a library management system's components operate in concert. Each of these components makes up a subsystem, which itself is made up of a number of related components. In other words, each of these subsystems may be divided into even smaller subsystems, such as a subsystem that manages the options for catalog display. It is noteworthy that the primary subsystems, such as acquisitions and circulations, are frequently referred to as modules by suppliers of library management systems.

An information system example is a library management system. A system that represents items in a physical system, such as information resources in a library collection, is an information system, whether it is computerized or not. A system that represents the real information resources of a library, whether that representation consists of markings on a card, marks on a microfiche sheet, or data saved on a computer, is the catalogue outlined in the previous chapter.

3. METHODOLOGY

It is uncommon to have a Java application that is both compiled and interpreted. A program is initially converted into Java byte codes using the compiler; these codes are platform neutral and are then decoded by the Java platform interpreter. The computer's interpreter parses and executes each Java byte code instruction. Java byte codes can be compared to the Java Virtual Machine's machine code instructions (Java VM). Every Java interpreter, whether it is a programming tool or a Web browser that supports applets, is an implementation of the Java Virtual Machine. It is possible to "write once, run anywhere" thanks to Java byte codes. Your program may be converted into byte codes on any platform that has a Java compiler.

ODBC

A common programming interface for database system providers and application developers is Microsoft Open Database Connectivity (ODBC). Before ODBC became the de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages to connect to any database they desired. The database system is now practically irrelevant from a coding perspective thanks to ODBC, which is exactly how it should be.

From a programming perspective, ODBC is advantageous because it enables an application to use the same set of function calls to connect with any data source, irrespective of database provider.

The source code is the same whether the software interacts with Oracle or SQL Server. These are but a few illustrations. There are ODBC drivers available for several popular database systems. Plain text files and Excel spreadsheets can be used to construct data sources.

JDBC

To provide a separate database standard API for Java, Sun Microsystems established Java Database Connectivity, or JDBC. A common interface to a variety of relational database management systems is provided by JDBC, a standard SQL database access technology (RDBMSs). A standard interface is made possible by the use of "plug-in" database connectivity modules or drivers.

The driver must be available for each platform that Java and the database are supported on if a database vendor wants JDBC support. Sun based the design of JDBC on ODBC to increase adoption. As you already know from this chapter, ODBC is extensively supported by a wide range of platforms.

API at SQL Level

The designers considered creating a Java SQL interface to be the most crucial challenge. It's not the lowest database interface level possible, but it's low enough to let higher-level tools and APIs be developed. On the other hand, it is sufficiently advanced for application programmers to utilize it safely. Future tool developers will be able to "generate" JDBC code and hide much of JDBC's complexity from the end user in order to accomplish this goal. The SQL syntax varies when you go from one database supplier to another. To support a wide range of vendors, JDBC will enable any query statement to be sent via it to the underlying database driver. As a result, non-standard capabilities can be handled by the connection module in a user-friendly manner.

The suggested design is an automated library management system. Users may add members, add books, search members, search books, update information, edit information, borrow books, and return them fast thanks to the program. Some advantages of the system we propose are listed below.

The interface is simple to use, and database access is speedy.

- Greater Storage and Lower Error Capacity
- Visual and tactile environment
- The transaction was finished fast.
- The Library may now be operated without encountering any manual challenges thanks to the adoption of computerization.

A well-liked method for creating interactive online applications is Java Servlets, which have essentially superseded the use of CGI scripts. Similar to applets, servlets are program runtime extensions. As opposed to browsers, servlets work within Java Web servers, configuring or changing the server.

How do all of these programs get support from the API? This is achieved by using software component packages that offer a wide range of functionality.

The following characteristics are included in a full Java platform implementation:

Basic ideas include things like objects, threads, integers, strings, input and output, data structures, system properties, date and time, and so on.

Applets: The collection of practices used by applets.

A wide range of relational databases may be accessed consistently thanks to Java Database Connectivity (JDBCTM).

Internet Protocol (IP) addresses, TCP (Transmission Control Protocol), UDP (User Datagram Protocol), and URLs are all used in networking.

Help in developing programs that may be localized for users throughout the world is known as internationalization. Programs can automatically display in the appropriate language and adapt to specific locales.

Security: Both low-level and high-level security capabilities, including certifications, access control, public and private key management, electronic signatures, and public and private key management.

Components of software: Also referred to as JavaBeans TM, these parts may be inserted into already-existing component architectures.

Through Remote Method Invocation, object serialization enables lightweight persistence and communication (RMI).

4. RESULT & DISCUSSION

implementation phase involves careful planning, investigation of the present system and its limitations on implementation, development of changeover procedures, and evaluation of changeover approaches.

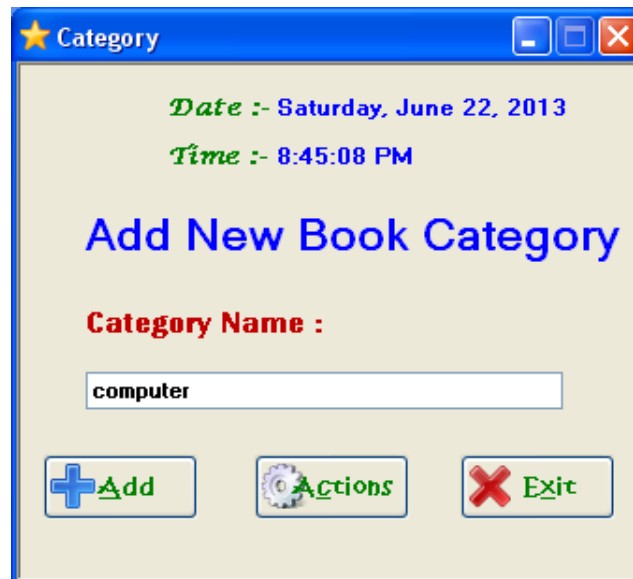


Fig 1. Adding book category

Add/Remove Books

Date :- Saturday, June 22, 2013 Time :- 8:46:23 PM

Add Books Remove Books

Add Books

<i>Book Title</i>	C
<i>Author Name</i>	Brooklyn
<i>Category Name</i>	ComputerScience
<i>Publisher</i>	NV
<i>Place Of Publish</i>	Delhi
<i>Year Of Publish</i>	2001
<i>Volumes</i>	4
<i>Edition</i>	2nd
<i>Pages</i>	768
<i>Price</i>	590
<i>Quantity</i>	5

+ Add

⚙ Actions

✖ Exit

Fig.2 Add or remove books

The process of turning a theoretical design into a practical system is known as the implementation stage of a proposal. It may thus be considered the most important phase in assuring the success of a new system and giving the user confidence that the system will work and be efficient.

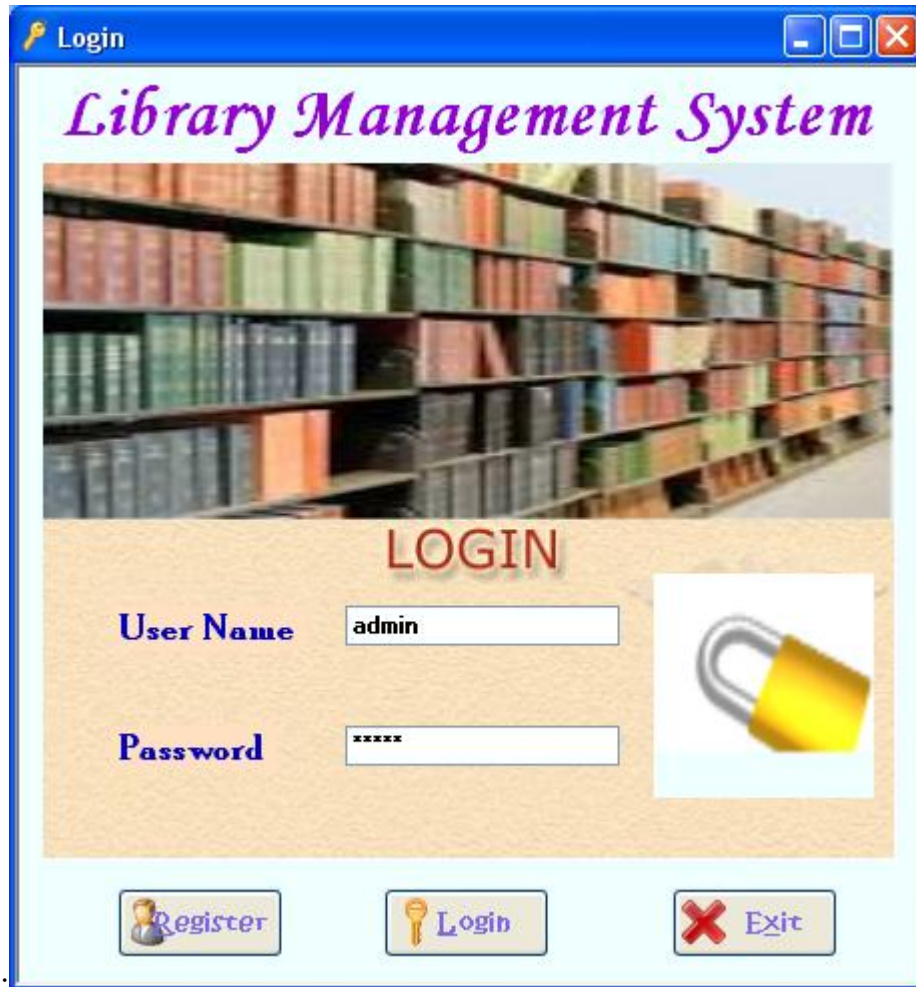


Fig 3. Login page

The major goal of maintenance is to lessen or eliminate the consequences of equipment failure. This might be achieved by planned maintenance or condition-based maintenance, which both aim to stop failures before they start. By replacing worn components before they fail, it is meant to protect and restore equipment reliability.

Oil changes, lubrication, and other related duties are examples of preventive maintenance procedures, which also involve partial or complete overhauls at regular intervals. Workers may also keep track of equipment deterioration in order to replace or fix damaged components before they lead to system failure. All equipment faults would be found and prevented by the perfect preventive maintenance program.

5. CONCLUSION

Our suggestion merely makes a merger effort to satisfy a library's needs. There are also a number of user-friendly coding methods in use. This package will prove to be an excellent

instrument for satisfying all of the demands of the company. At the beginning of the software project, software planning aims to provide a framework for the manager's development so they can quickly create accurate projections. As the project moves forward, this framework should be revised often.

6. REFERENCES

- [1] Adegboye, A. M. (2010). Automation in two Nigerian university libraries.
- [2] Boss, R.W. (2008), “‘Open source’ integrated library system software”, available at: www.ala.org/ala/pla/plapubs/technotes/OpenSource.doc (accessed 21 February 2009).
- [3] De Smet, E. (2008). The ISIS-software family: from ‘Free and Open ‘to ‘Free and Open Source Software’. *Innovation*, 36(1).
- [4] De Smet, E. (2010). Some ISIS-Software history and technical background on the new FOSS integrated library system ABCD. *Liber Quarterly*, 19(3/4), 324-335.
- [5] De Smet, E. (2010). Some ISIS-Software history and technical background on the new FOSS integrated library system ABCD. *Liber Quarterly*, 19(3/4), 324-335.
- [6] Husain, S. and Ansari, M.A. (2007), “Library automation software packages in India: a study of cataloguing module of Alice for Windows, LibSys and Virtua”, *Annals of Library and Information Science*, Vol. 54 No. 3, pp. 146-51.
- [7] Jibia, M. S., Mubaraka, C. M., & Michael, O. (2013). Integrating ICT in Library Management: Design and Development of an Automated Library Management System for Cavendish University Uganda. *Innovative Systems Design and Engineering*, 4(5), 1-11.
- [8] Mukhopadhyay, P. (2002), “Progress of library management software: an Indian scenario”, *Vidyasagar University Journal of Library and Information Science*, pp. 1-15.
- [9] Ravichandra Rao, I.K. and Abideen Sainul, P. (1999), “Features of library automation software: a comparative study: library science with a slant”, *Documentation and Information Studies*, Vol. 36 No. 4, pp. 211-28.