Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

1999

Web-Based Interlibrary Loan System

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

Schubert FOO
Nanyang Technological University, Singapore

Cheng Hai TAN

Kwang-Yong TAN

Citation

LIM, Ee Peng; FOO, Schubert; TAN, Cheng Hai; and TAN, Kwang-Yong. Web-Based Interlibrary Loan System. (1999). *Network Information: Official Journal of the Library Information Technology Association*. 3, 15-34. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/746

Web-Based Interlibrary Loan System

Ee-Peng Lim, Schubert Foo, Cheng-Hai Tan, Kwang-Yong Tan
School of Applied Science
Nanyang Technological University
Nanyang Avenue, N4-2A-36
Singapore 639798

ABSTRACT

Interlibrary loan (ILL) system is one of the important library services provided by most public libraries. Through ILL, public libraries share their collections of reading material and library users are able to access library resources beyond their affiliated libraries. However, the present manual ILL process is inconvenient, inefficient and error-prone. In this project, we developed a system that automated many tedious procedures in the existing ILL system. The new ILL system is realized by a number of Java applets customized for librarians and library users. The new ILL system allows library users to submit ILL requests at any time using their favorite Web browsers without being physically present at the libraries. Librarians can also process ILL requests as and when they have accessed to Java-enabled Web browsers. The new ILL implementation is achieved without forcing libraries to sacrifice their local autonomy over their legacy library systems, e.g. Online Public Acess Catalogue (OPAC) System. For each participating library, two databases containing incoming and outgoing ILL requests respectively are maintained. The entire ILL process is tracked by multiple ILL Java applets updating the borrowing and lending libraries' databases.

Keywords: Interlibrary loan, digital libraries, World-Wide Web

1. INTRODUCTION

Libraries today offer a wide variety of services that cater to different types of users. Among these services, Inter-Library Loan (ILL) service allows users affiliated with a library to request for books or library items from the other libraries when these library items cannot be found in their affiliated library. Such a service widens the range of library material offered by a library and also optimises resource usage among libraries participating in ILL. The current ILL procedures, which are performed manually by the library staff, are both inefficient and error-prone. The librarian has to manually fill in various forms, source for the requested items, submit the request and coordinate the collection of items. In the Singapore context, it takes an average of six man hours just to get the approval from the lending library. The collection, tracking and controlling of loaned material incur even more man hours. It has been found that much ILL processing time is spent in form completion and updating of loans' statuses that require minimal decision making. At the Nanyang Technological University Library[1], a full-time staff has been assigned to process ILL requests.

To reduce ILL delays and to minimize manual effort, there is an urgent need to automate the current ILL procedures. The automation will not only improve the response time, but also allow more library staff to be re-deployed in other library operations. In this project, we aim to automate the ILL process making it easy to use for both librarians and library users. As part of an overall effort to make libraries widely accessible from the World Wide Web (WWW), we would like to operate the new ILL system using the popular Web browsers, e.g. Netscape Communicator[2] and Microsoft Internet Explorer[3].

1.1 Pitfalls of the Existing ILL System

Interlibrary loan services in Singapore are currently provided by librarians processing the ILL requests manually. Apart from using OPAC to look for books within a library, most of the ILL procedures require manually making telephone calls to different libraries, checking the availability of requested items, monitoring the progress of ILL requests, and informing users to collect or return their request items. Interaction between the library and users, and between libraries, usually involves many forms, and incurs much overheads. During the user requirement phase of this ILL project, a brief survey was conducted with the librarians and users and it revealed the following problems frequently occur in ILL:

- It is time consuming to attend to telephone enquiries about ILL requests.
- It is tedious to fill up the various ILL request forms. It was found that different ILL request forms often require the same information to be supplied at different points in time. The ILL process could be simplified if the ILL information are captured in some database and are reused when necessary.
- Manual filing of request forms and statistical collection are time consuming.
- The communication problems between the lending and borrowing libraries often cause delay in processing ILL requests.
- Much time is wasted trying to contact library users.
- The current ILL services are not fully utilized by the library users possibly due to the overheads.

1.2 Objectives

The main objectives of the new ILL system include:

- Developing an ILL system for multiple libraries in the education institutions. Based on the current ILL procedures practised by Nanyang Technological University Library, the new ILL system is expected to be deployed in junior colleges, polytechnics and universities in Singapore.

- Allowing librarians to process ILL requests from anywhere as long as they have access to machines with Java-enabled Web browsers. Similarly, library users can submit ILL requests from home PCs or any other machines running Web browsers. Hence, the new ILL system is ubiquitous.

- Allowing both librarians and library users to monitor the status of ILL requests easily without going through stacks of ILL forms. Hence, the new ILL system can relieve the existing librarians from a lot of paperwork and monitoring overhead.

- The new ILL system is non-intrusive: it should not require the existing libraries to give up their existing library systems. In fact, the libraries participating in the ILL system should be able to maintain their ILL databases conveniently. They should also be allowed to out-source the ILL operations and database maintenance to third parties who are able to provide the support more cost effectively and efficiently.

- The design and implementation of the new ILL system must be open enough for future enhancement. To cope with the rapid advancement of Web technologies, an open design will permit our ILL system to be upgraded easily.

1.3 Outline of Paper

The rest of this article is structured as follows. Section 2 gives an overview of the new ILL system. Section 3 describes the workflow to be performed in the new ILL system. Some important user interface design is presented in Section 4. Conclusions and future works are described in Sectio n 5.

2. OVERVIEW OF NEW ILL SYSTEM

In this section, we give an overall description of the new ILL system including its system architecture and the functionalities of various system components.

2.1 System Architecture

The new ILL system adopts a client-server architecture as shown in Figure 1. The ILL client software executes on the PCs or workstations at the user sites. Each library is equipped with a set of ILL server software. The ILL server software consists of a *SQL database server* and an *ILL gateway.* A library usually functions as borrowing and lending library at the same time. Hence, the same architecture design is adopted for both borrowing and lending libraries.

The client software consists of Java applets[4] that run in Netscape Navigator or Microsoft Internet Explorer on PCs or workstations connected to the Internet. Applets are essentially portable computer programs that can run on Netscape and Internet Explorer browsers. A Java applet is always stored at a Web server and is only loaded into a client computer just before its execution. Hence, no prior installation of applets at the client computer is necessary. While one of the applets provides the ILL frontend for a librarian to manage ILL requests, the other provides the frontend for a library user to make their ILL requests. The ILL Java applets for the librarian and library user can be accessed via the ILL librarian Web page and the ILL library user Web page respectively. When a Web browser is used to view these Web pages, both the Web pages and their corresponding ILL Java applets will be loaded from the Web server onto the client computer.

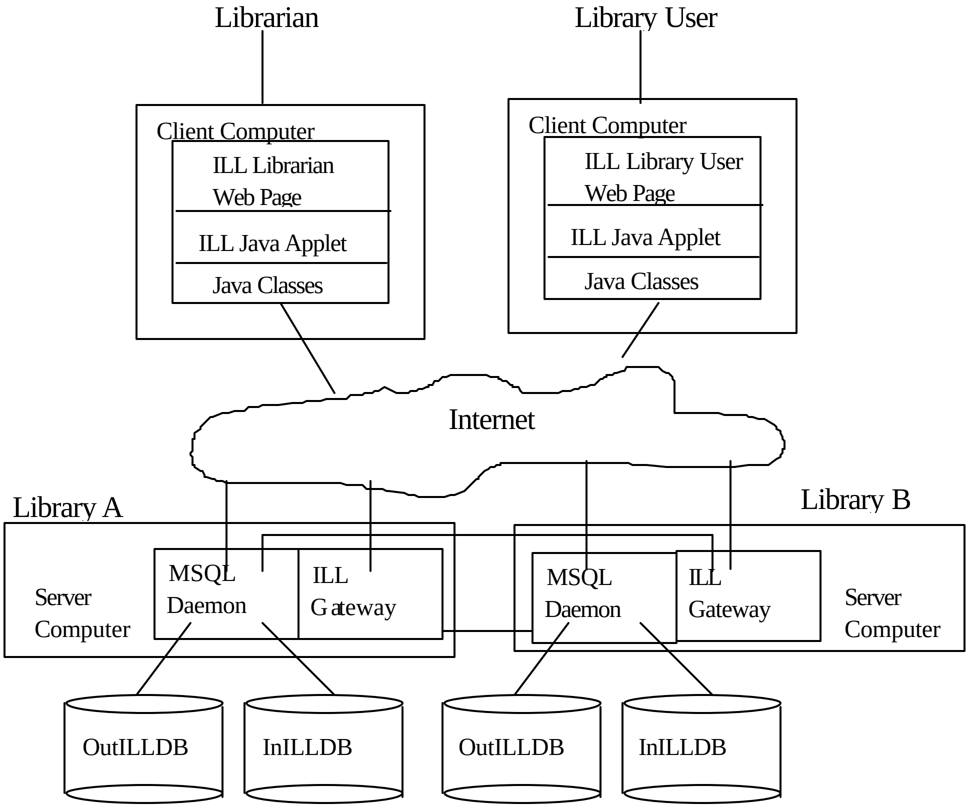


Figure 1: Overall System Architecture

On each server computer owned by a participating library, one can find two ILL databases: one storing the incoming ILL requests and another storing the outgoing ILL requests. The incoming ILL requests contain items required by library users from the other libraries. The outgoing ILL requests contain items required by the local library users. In other words, all outgoing ILL requests of a borrowing library are stored in its **OutILLDB** database. When an ILL request is forwarded to a lending library, the **InILLDB** database of the lending library will be updated with the ILL request. In this project, both databases have been implemented using a SQL database package known as MiniSQL[5].

As librarians and library users interact with the ILL Java applets, the two ILL databases have to be queried or updated. To query or update a database, an ILL Java applet invokes methods that connect and transact with the database. Java-mSQL API (part of MiniSQL) [6] provides the necessary Java classes required by the ILL applets to perform database queries and updates.

To process an ILL request at a borrowing library, the OutILLDB database of the borrowing library will be updated. The request also causes updates on the InILLDB database of the lending library. When the librarian of the lending library responds to the ILL request, the OutILLDB database of the borrowing library will be updated. Since a Java applet is only allowed to communicate with its host Web server, it cannot directly manipulate a database that resides on a Web server different from its host Web server. To ovecome this security restriction imposed by Java applets, an ILL gateway of the borrowing library server is responsible for relaying database queries and updates to the lending library server and updating the InILLDB database of lending library. The ILL gateway performs the queries and updates on the lending library by establishing direct connection to the mSQL server of lending library on behalf of the borrowing library

3.2 ILL Database Design

Two ILL databases (i.e. InILLDB and OutILLDB) are maintained for each library participating in ILL. The former is required for keeping information about incoming ILL requests and the latter for outgoing ILL requests. The information contained in the OutILLDB database is summarized below.

a. All ILL requests are uniquely identified by their system generated request serial number.

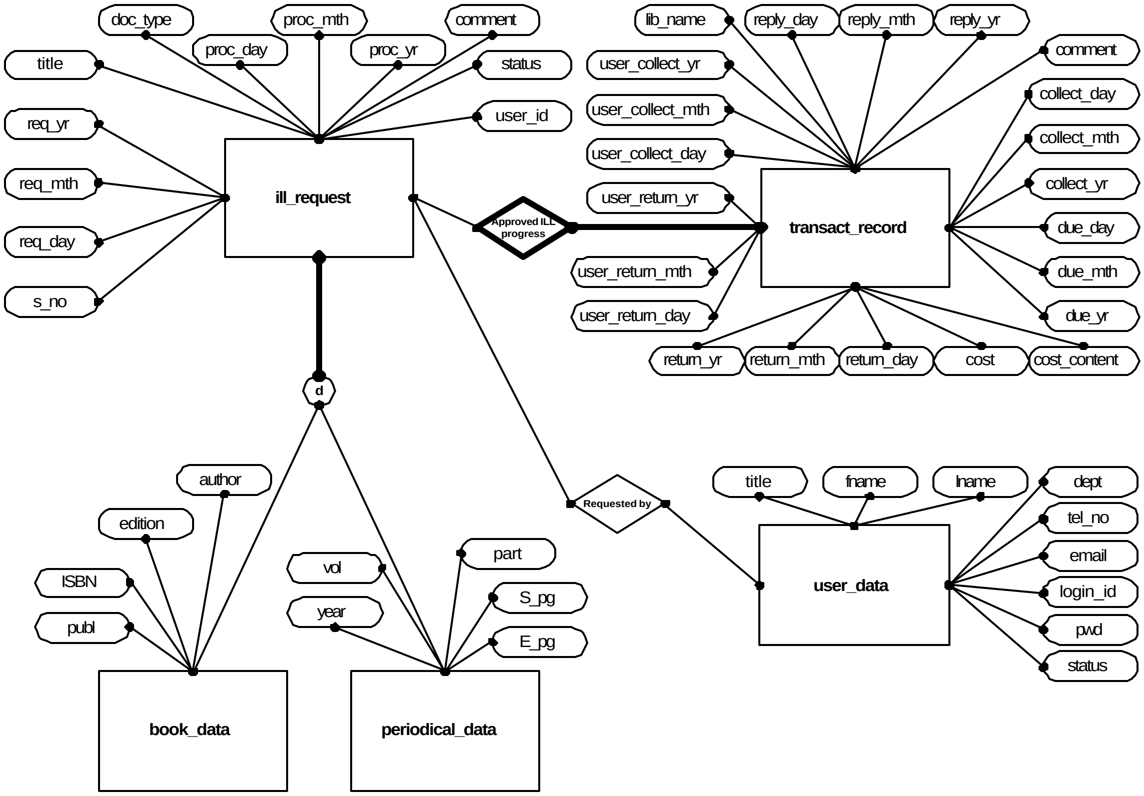
b. The list of attributes of an ILL request include the request submission date, title of item, item type, library user id, request processing date, comment on the outcome and current status.

c. The information maintained for each authorized library user include the user identifier, title, first name, last name, department, contact telephone number, email address, login name, password and authorization status. Library users are uniquely identified by their user ids.

d. Loan items are either books or periodicals. The essential information of book items include the book title, author, edition, ISBN number and publisher. On the other hand, for each periodical item, the information to be kept include the periodical title, year, volume number, part number, start and end page numbers.

f. All approved loans are tracked by monitoring transactions throughout the cycle. For each ILL request, we maintain a transaction record in the database. Each transaction record consists of lending library name, reply date, reply comment, collection date, due date, user collection date, user return date, library return date, cost and cost content. Each transaction record is related to its ILL request by the request's serial number.

The above information requirement has been modeled by an Entity-Relationship (ER) diagram[7] depicted in Figure 2.



Note: Due to limitations of mSQL database, date's attribute are separated as Day, Month and Year.
BOLD line denotes TOTAL participation and BOLD diamond denotes DEPENDANCY.

Figure 2: ER Diagram for Outgoing ILL Request Database (OutILLDB)

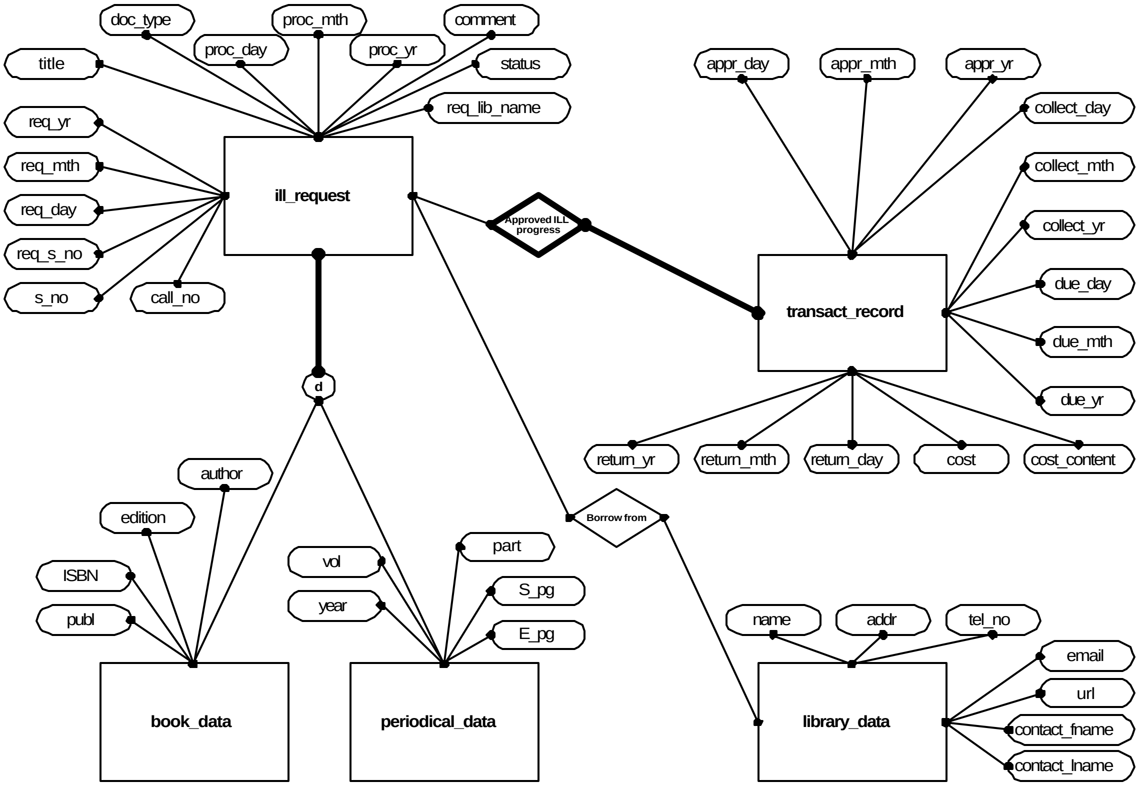
The design of InILLDB is quite similar to that of OutILLDB. However, due to the nature of incoming ILL requests, we need to capture different sets of attributes for the incoming ILL requests and their transaction records. Moreover, instead of keeping information about library users who make the ILL requests, we now maintain information about the borrowing libraries. These information are summarized below.

a. For each ILL request, we need to keep the serial number of the ILL request issued by the borrowing library and the identity of borrowing library. The user id of the library user making the request is not required because the transaction is between the borrowing and lending libraries instead of between the user and the lending library. Using the above information, one can determine the borrowing library and the ILL request originated from the borrowing library.

b. Similar to that in the OutILLDB, we need to maintain a transaction record for each incoming ILL request. However, the transaction record only maintains various important date information including the date of approval by lending librarian, date of collection, due date for the library item, and the date of returning the item.

c. Since the ILL transactions are carried out directly among libraries, we need to maintain information about the participating libraries. Such information include the names of different libraries, their addresses, contact telephone numbers, email addresses, the URLs of their library homepages, the first and last names of the contact persons.

The Entity-Relationship diagram for modeling information contained by the InILLDB is shown in Figure 3.



Note: Due to limitations of mSQL database, date's attribute are separated as Day, Month and Year.
BOLD line denotes TOTAL participation and BOLD diamond denotes DEPENDANCY.

Figure 3: ER Diagram for Incoming ILL Request Database (InILLDB)

3. THE WORKFLOW OF NEW ILL SYSTEM

Based on the existing ILL procedures, we design the workflow of the new ILL system. Since the new ILL procedures involve both the borrowing and lending libraries, we will devote the following two subsections to the ILL activities in both the borrowing and lending libraries.

3.1 ILL Workflow at the Borrowing Library

The ILL workflow at the borrowing library consists of two main activities, i.e. *user registration* and *outgoing ILL request processing.* Prior to any ILL request, a library user must first register with the ILL system. Figure 4 depicts the user registration process carried out by a library us er. A user begins by filling an online user registration form on the Web. The registration form will then be approved by the librarian. Users can check the registration Web page sometime after their registration in order to find out if their registratio n have been approved.

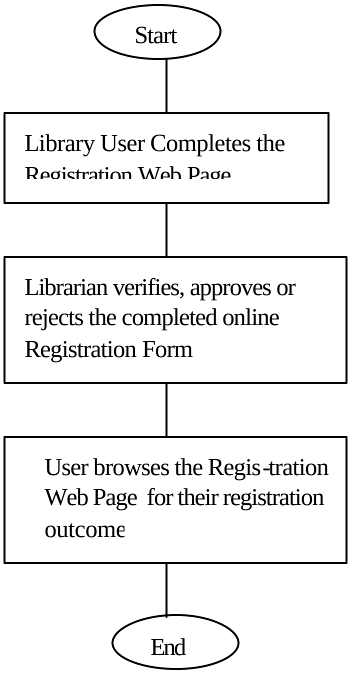

Start
Library User Completes the Registration Web Page
Librarian verifies, approves or rejects the completed online Registration Form
User browses the Regis-tration Web Page for their registration outcome
End

Figure 4: User Registration Activity

A registered user can submit an ILL request by filling in an ILL request form using the ILL Java applet in the ILL Library User Web Page. Such request will be stored into the OutILLDB and be verified by a librarian of the affiliated library which is also known as the borrowing library. The verification is performed using the ILL Java applet in the ILL Librarian Web Page. The librarian of the borrowing library will approve or disapprove the request. Upon approval of an ILL request, the librarian has to source the holdings of the requested library item by checking the Union Catalogue constructed for collections owned by libraries in Singapore (only applicable to libraries which are members of the Union Catalogue) or Web interfaces to other libraries. The lending library will be identified and its InILLDB database is updated with the request. The borrowing library will also update the request status in its OutILLDB database.

If the ILL request is rejected by the librarian of the borrowing library, the status attribute of the rejected ILL request record in the OutILLDB database is assigned an unsuccessful status. The user can find out that the request is rejected when he/she checks the request status. On the other hand, an approved ILL request is forwarded to the lending library. The librarian of the lending library will subsequently process the ILL request as described in Section 4.2. When the lending library approves the request, the approval is recorded with the OutILLDB database of the borrowing library. The librarian of the borrowing library will be able to monitor the progress of ILL request and arrange the requested item to be collected from the lending library. Once an item is collected, the status of the ILL request in the OutILLDB database is further updated. When the user collects the item from the borrowing library, the status of the ILL request in the OutILLDB database is updated again. When the user returns the item, the borrowing library updates the request status in its OutILLDB database and returns the item back to the lending library. The lending library will deal with the returned item as described in Section 3.2. The processing of outgoing ILL request is illustrated in Figure 5.

3.2 ILL Workflow at the Lending Library

There are two main ILL activities to be carried out at the lending library, namely *processing of incoming ILL requests* and *processing of return items*. New incoming ILL requests from the borrowing libraries are first recorded in the InILLDB database of the lending library. By using the ILL Java applet embedded in the Librarian Web Page, a librarian can find out all incoming ILL requests to be processed. By checking the availability of the requested item using OPAC, the librarian can determine if a request can be approved. Approval or rejection of the ILL request will trigger update to the status of the corresponding ILL request record in the InILLDB database. In order to inform the borrowing library about the status, the ILL request record in the OutILLDB database of the borrowing library is updated as well. When the requested item is collected by the borrowing library, the ILL request record in both the InILLDB database of the lending library and the OutILLDB database of the borrowing library are updated to reflect the latest status. This entire activity is shown as a flowchart in Figure 6.

The processing of a return library item begins when the borrowing library returns a library item to the lending library. One first determine the ILL request record corresponding to the returned item and update the ILL request record accordingly. If the return is overdue, some monetary penalty may optionally be imposed on the borrower. The processing of a return library item is depicted in Figure 7.

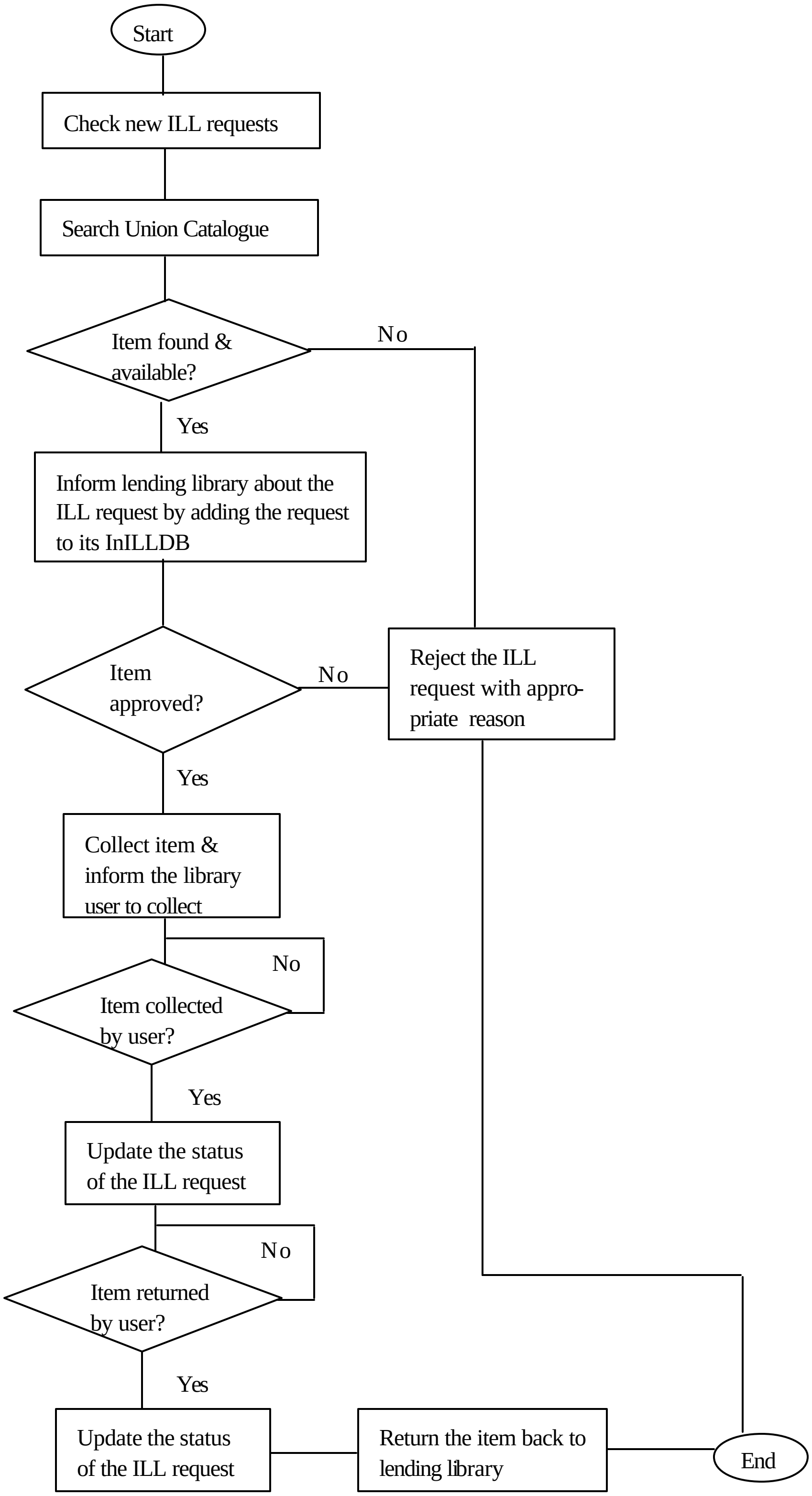

Start
Check new ILL requests
Search Union Catalogue
Item found & available?
No
Yes
Inform lending library about the ILL request by adding the request to its InILLDB
Item approved?
No
Reject the ILL request with appro-priate reason
Yes
Collect item & inform the library user to collect
No
Item collected by user?
Yes
Update the status of the ILL request
No
Item returned by user?
Yes
Update the status of the ILL request
Return the item back to lending library
End

Figure 5: Processing of An Outgoing ILL Request

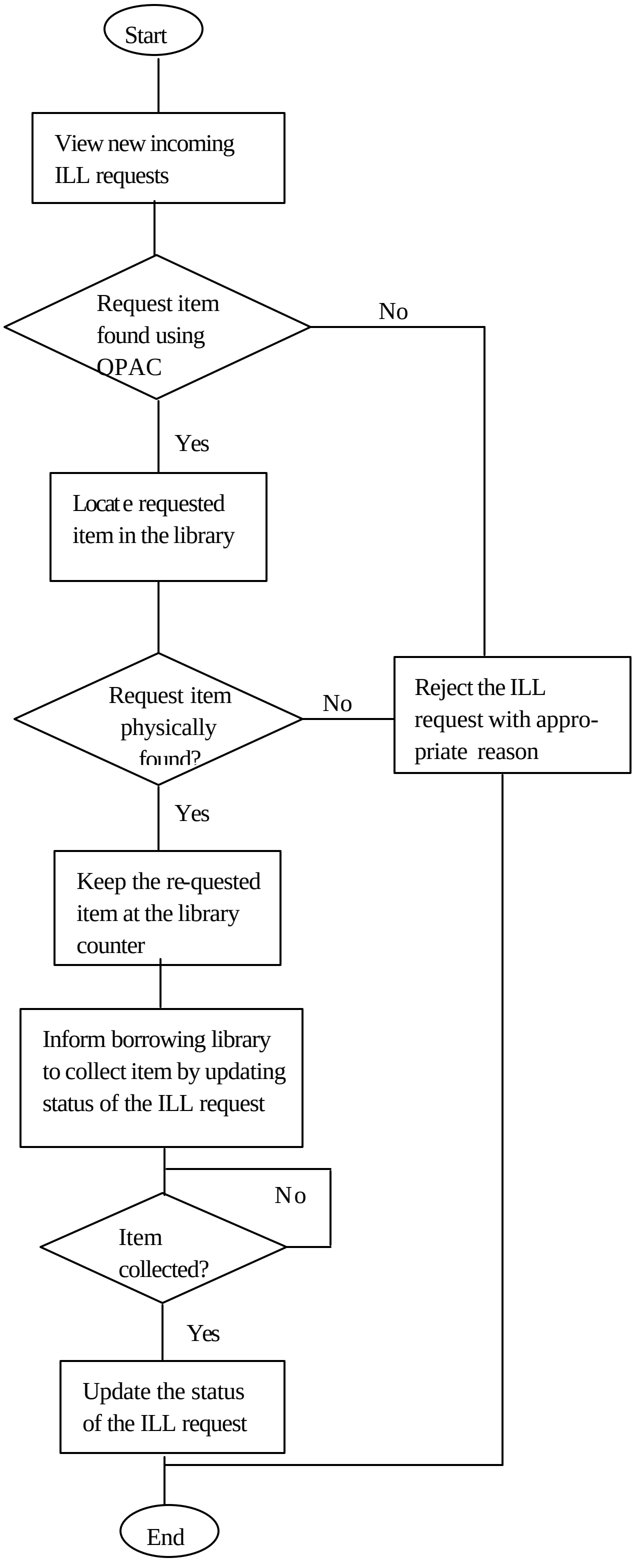

Start
View new incoming ILL requests
Request item found using OPAC
No
Yes
Locat e requested item in the library
Request item physically found?
No
Reject the ILL request with appro-priate reason
Yes
Keep the re-quested item at the library counter
Inform borrowing library to collect item by updating status of the ILL request
No
Item collected?
Yes
Update the status of the ILL request
End

Figure 6: Processing of an Incoming ILL Request

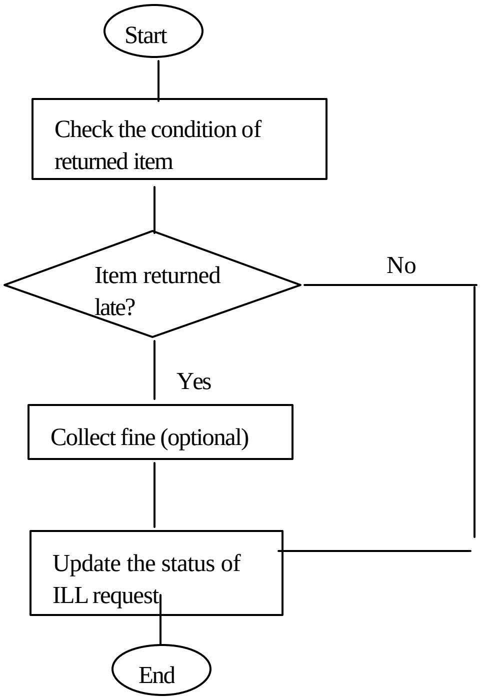

Start
Check the condition of returned item
Item returned late?
No
Yes
Collect fine (optional)
Update the status of ILL request
End

Figure 7: Processing of a Return Library Item

4. USER INTERFACE DESIGN

The new ILL system is equipped with a set of user-friendly interfaces implemented mostly as Java applets embedded in the Librarian Web page and the Library User Web page that are accessed by the librarian and library users respectively. In Figure 8, we show the user registration electronic form to be completed by library users before they make their first ILL requests. As shown in Figure 9, a Librarian Web page requires users to provide their login names and passwords prior to other ILL related operations. To keep the user interface generic and simple, a single Librarian Web page is designed for processing incoming and outgoing ILL requests. A librarian can also process the membership registration using the same Web page. Figure 10 depicts the Library User Web page accessed by library users to make new ILL requests, or monitor the progress of existing ILL requests. All user interface modules have been designed to minimize user input. By directly retrieving ILL request information in the InILLDB and OutILLDB databases in the library server, the user interface modules effectively remove redundant data entries by the librarians and library users.

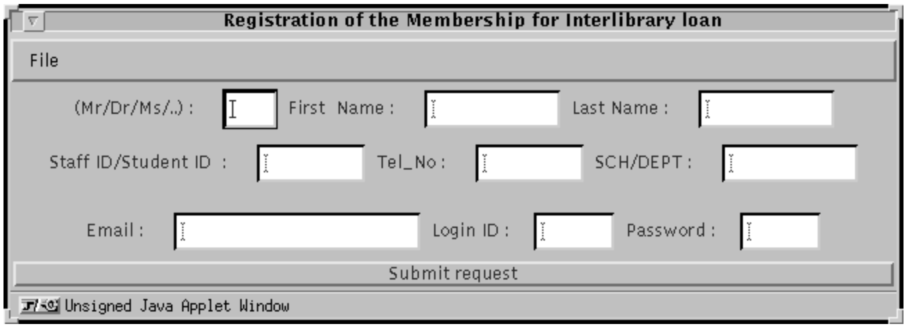
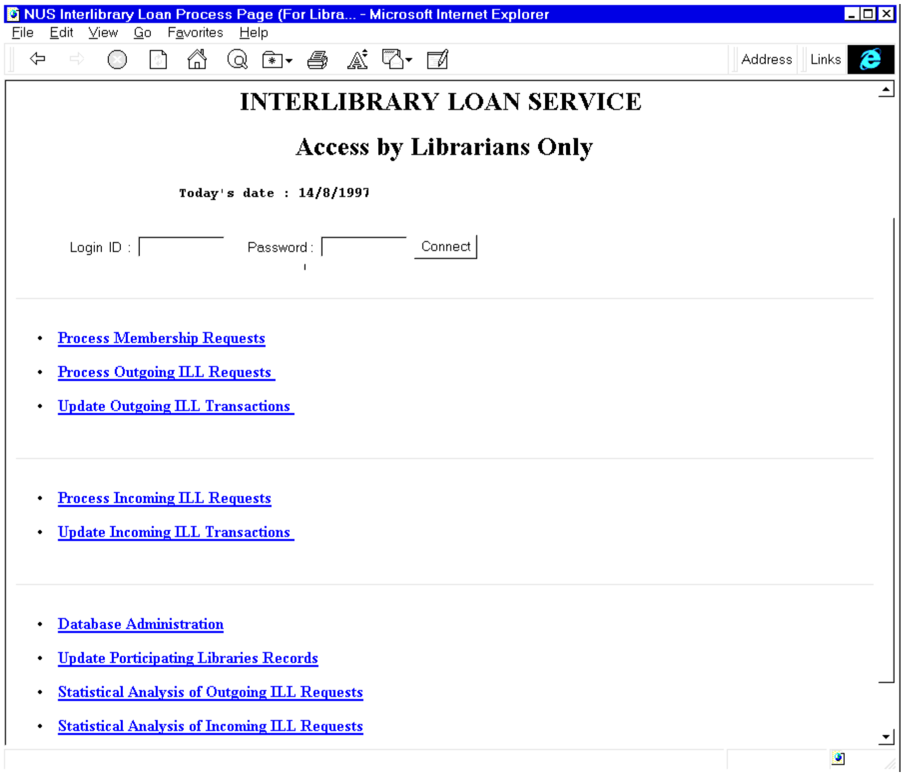


Figure 8: User Registration


NUS Interlibrary Loan Process Page (For Libra... - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address Links

INTERLIBRARY LOAN SERVICE

Access by Librarians Only

Today's date : 14/8/1997

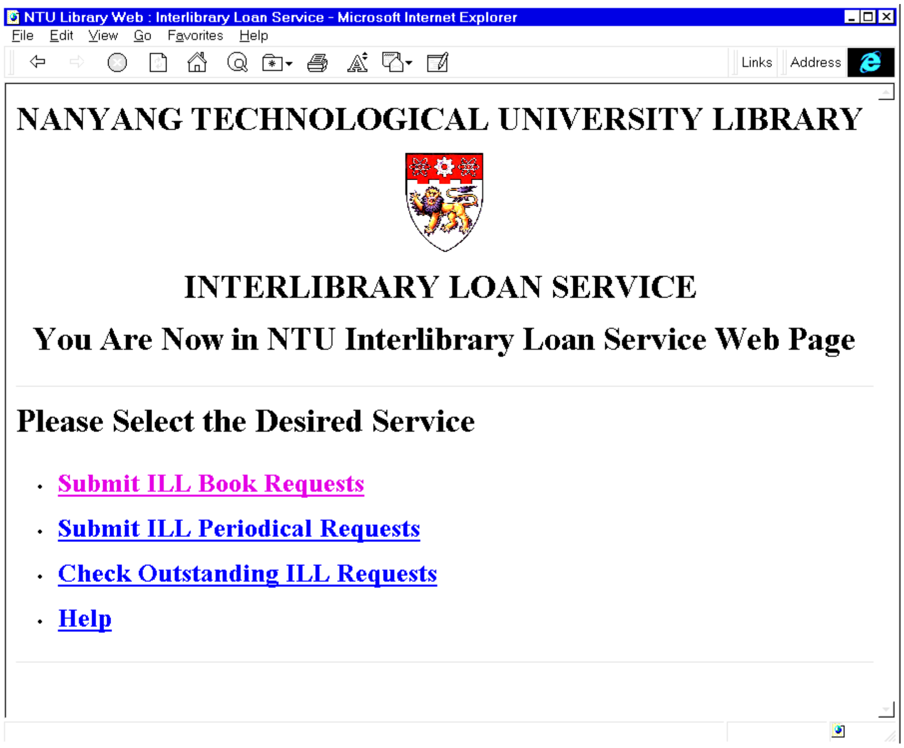
Login ID : Password : Connect

- Process Membership Requests
- Process Outgoing ILL Requests
- Update Outgoing ILL Transactions

- Process Incoming ILL Requests
- Update Incoming ILL Transactions

- Database Administration
- Update Participating Libraries Records
- Statistical Analysis of Outgoing ILL Requests
- Statistical Analysis of Incoming ILL Requests

Figure 9: Librarian's Web Page


NTU Library Web : Interlibrary Loan Service - Microsoft Internet Explorer
File
Edit
View
Go
Favorites
Help
Links
Address

NANYANG TECHNOLOGICAL UNIVERSITY LIBRARY

INTERLIBRARY LOAN SERVICE

You Are Now in NTU Interlibrary Loan Service Web Page

Please Select the Desired Service

- Submit ILL Book Requests
- Submit ILL Periodical Requests
- Check Outstanding ILL Requests
- Help

Figure 10: Library User Web Page

5. CONCLUSIONS

The objective of this ILL project is to automate the current manual ILL process so that they can be performed on the Web using the Java technology. In this section, we summarize the contribution of this project, point out the system's limitations and describe the possible future enhancements to the system.

5.1 Summary of Contribution

We summarize the contribution of this ILL project as follows:

- We have investigated the problems of the existing manual ILL system and developed a new approach to perform ILL on the Web. By operating ILL on the Web, we make ILL widely and easily accessible. The new solution is also cheap because it does not require libraries to purchase expensive computers or networks.

- The operation of our new ILL solution is very flexible. Participating libraries can choose to manage the ILL databases and programs by themselves or to engage third parties to operate the ILL databases and programs. Hence, our ILL solution satisfies the requirement of libraries with strong information system support as well as libraries without information system personnel.

- The new ILL solution upholds the autonomy of pre-existing library systems. Pre-existing library systems are heterogeneous and it is difficult to migrate library databases and programs into new systems in a short time. The new ILL solution manages to provide additional functionality to the existing library systems without intruding these heterogeneous systems.

- The new ILL is easily scalable. There is no central repository of ILL information for all participating libraries in our solution. Each library has its databases thus avoiding data replication and single performance bottleneck. This approach is different from the upcoming ILL solution provided by TIARA[1] which requires a central database to store all ILL information. In the TIARA's ILL solution, each library is required to contribute their catalogue information to a central database regularly in order to keep the central database up-to-date.

To verify the feasibility of the proposed Web-based ILL system, a prototype of the new ILL system has been developed. The initial feedback from the librarians was positive and encouraging. Hence, we plan to experiment the ILL system within Nanyang Technological University, using the libraries physically located in two campuses (i.e. Yunan Garden and Bukit Timah campuses) as test sites.

5.2 System Limitations

At present, there are some limitations of the new ILL system:

- As MiniSQL database package does not support transaction management, concurrent accesses to the database tables may be inconsistent due to incorrect interleaved sequence of database operations. To overcome this limitation, we need to replace the MiniSQL database package by a proper SQL engine supporting JODBC.

- The ILL Java applets when run on different platforms may look differently due to different implementation of the basic Java classes by different vendors. For example, certain buttons may not be appear on the screen. Such deficiency can be avoided if only a fixed combination of client computer and Web browser is allowed.

5.3 Future Work

As the first project of its kind on the Web, our ILL system can be further enhanced in a number of ways.

[1] TIARA is the name of a digital library project under the Singapore's Library 2000 plan.

- At present, our solution does not integrate ILL activities with the other library functions. To improve its usability, we plan to integrate the ILL system with the existing OPAC systems so that OPAC query results can be used directly when users submit their ILL requests.

- To help librarians in maintaining the statistics of the use of ILL, we plan to develop a statistics module to collect the usage information of ILL services.

- Currently, library users are responsible to check the status of their ILL requests since our ILL system, in the current version, is not able to send notification emails to them. We therefore plan to incorporate auto-email service into our ILL system so that librarians and library users can be informed on the progress or outcome of the loan.

- Since security is not the main objective of this project, we did not implement comprehensive security modules in the project. We plan to enhance the security features of the ILL system in the future.

REFERENCES

[1] Nanyang Technological University, "Nanyang Technological University Library," URL: http://www.ntu.edu.sg/library/lib.html, 1998.

[2] Netscape Communications Inc., "Netscape Navigator for 2.02 WWW Browser," URL: http://www.netscape.com/comprod, 1997.

[3] Microsoft Corporation, “Internet Explorer 3.0,” URL: http://www.microsoft.com/ie/ie.html, 1997.

[4] Walnum, C., "Java by Example," Que Corporation, 1996.

[5] Hughes, D.J., "Mini SQL: A Lightweight Database Engine Version 1.0.6," URL: http://bond.edu.au/People/bambi.html, 1995.

[6] Collins, D., "Java-mSQL APIs," URL: http://www.minmet.uq.oz.au/~darryl, 1997.

[7] Elmasri, R. and Navathe, S., "Fundamentals of Database Systems," 2nd Edition, Benjamin/Cummings 1994.

[8] National Computer Board, "Timely information for All, Relevant and Affordable (TiARA)," URL: http://sitiara.digilib.org.sg/main.tiara.