

By

V.K.J Jeevan

Assistant Librarian

Indian Institute of Technology, Kharagpur,

West Bengal 721 302

E-mail: vkjj@rediffmail.com

ABSTRACT

This paper describes the in-house software development initiative undertaken at the Central Library of Indian Institute of Technology, Kharagpur for library automation and to replace the existing commercial package. A glimpse of the application of software development strategies for a library software development work is presented with the help of ER diagram, DFD, and schematic diagram, for the Current Journal module. Any software development effort pursues its challenges as the focal point to improve, enhance and substantiate the development effort. Some of the hidden and perceived challenges are listed out. As the software development phases up to testing only consume 40% of the effort, the library is gathering resources and support to execute 60% of the effort in maintaining the package as well as to generalize the package for its marketability.

KEY WORDS: Library software development, Oracle, Journal Module, Library software design

0. INTRODUCTION

Engineering and technical institutions focus on applied research and the results of such research find immediate social applications. These Institutions in the country were in the forefront of using computer and information technology tools for qualitative improvement of the erstwhile manual operations for collecting, processing, organising and servicing information sources and in delivering different user-centred information services. The Indian Institute of Technology at Kharagpur, the first, biggest and most diversified among IITs, established in 1951, is a trend setter in training, research and consultancy in different branches of applied sciences and technology. The Central Library (CL) of the Institute is rendering a substantial role in satisfying the information requirements of the faculty, staff and students numbering more than 7000. The orientation of the library has been solely in the manual mode up to late 80's and has been a mix of manual and automated in the 90's and the new century has seen substantial reduction of manual procedures. Presently, the CL is one of the largest and fully automated technical libraries in the country with lot of resources available online, a modern electronic library, CD-ROM Networking as well as hard disc serving, and fiber optic (ATM) LAN with VSAT connectivity.

1. PAST

Though the CL commenced the use of computer systems in 1987, the automation activities gathered momentum only in 1993 [1]. Since then, catalogue details of a massive collection had been converted into computer readable form, using the facilities prevalent at different times using the Unix servers of the Institute or the CL. Initially data regarding 1.5 lakhs collection was converted using Unibase and another 1 lakh using Unirec, concentrating only on automating the issue/return operations. As the search and user support facilities offered by these data management systems were primitive, commercial software (hereafter referred as 'commercialware') was purchased. The data available then in Unibase and Unirec were converted into and the information about new books is added to the commercialware. Much headway in these activities was made through the Library Automation and Manpower Project (LAMP) taken up with the help of some research projects offered to CL [2]. All library house keeping operations like acquisition of books, their card printing, circulation etc. are done through the commercialware. The library Online Public Access Catalogue (OPAC) provides information about over 3 lakh books and other reading materials like bound journals, theses, standards, microforms, etc. Achieving such a target in 1994 was a unique feat even at the national level. Provision was kept to continue most of the operations in manual form in the event of erratic power supply. All issuable materials were barcoded and issue/return is done in a user&/staff-friendly manner with the help of barcode scanners.

2. PRESENT

Our experiences suggest that library automation software has to be purchased and used not like other application software or commercial bibliographic or full text databases. Mere procurement is the end of the story for many other software/databases used in a library where as it will open a floodgate of problems and suitable action plans to counter them from the library in the case of library automation package. What the vendor will provide you is a skeleton to feed information concerning your collection, user, and service information. Thus purchasing suitable software is a complicated process with lot of intricacies. Relevant software for a particular library has to be evaluated from various angles such as, suitability of the software for the target library, its existing clients, hardware and operating systems supported computer awareness among staff and users, network as well as computer infrastructure in the Institution, and above all the cost and after-sales support. Any commercial ware for that matter developed for a general library (without any target library in consideration) has to be customized effectively for the individual library. This needs a lot of time, persuasion, and interaction from the library authorities and staff with the vendor. Also, automating a section's activities needs constant monitoring by the In-Charge of the section with active support of technical, data entry and system staff. In the CL, even after 7 years of working with the commercial ware, the penetration of complete automation has been quite mixed as customizing a commercial ware is an involving process. The book related sections of Acquisition, Processing and Circulation were better customized in the commercial ware whereas the journal sections found refuge in a mix of packages as customization of these section didn't realize much largely due to the software, administrative and personnel problems. Some details about the activities of different sections without using the commercial ware or through manual mode is presented in Table 1.

No.	Section	Manual/No-Commercialware Operations
1	Circulation	Issuing No-Dues, Generating user barcodes through another linux server
2	Acquisition	Still uses MS Word to generate Order and payment forms, letters, Payment part still done manually, manual accession register
3	Book Process (Technical)	Generation of spine labels manually
4	Reference	ILL operation manually
5	Periodicals (Current Journals)	Most of the operations are handled through dBase and MS Word. User search/retrieval services are managed through Web Site. Only the journal names are registered in commercialware for update by Back Volumes
6	Back Volumes (Old Journals)	For generating binding lists and other binding related tasks, dBase/Foxpro is used. Manual binding payment, accession register and spine labels.
7	Non Book	Manual accession register and spine labels of standards, theses, microforms etc.
8	Electronic Library	Manual Inventory register for CDs, floppies, videocassettes, Manual Usage register
9	SC/ST	Manual Accession Register, Spine labels

10	TBL (Text Book Library)	Manual issue/return for reserved collection
11	Automation	Manual inventory register, follow-up of system problems

Table - 1: Manual/Non-Commercialware operations in the CL

3. WHY IN-HOUSE DEVELOPMENT ?

Though the CL has been working with the commercialware for several years, attempts at customization continued as an ongoing process, the existing system is found to have inadequate functionality to incorporate many of the requirements. Similar is the case with most of the available packages as most of such packages have been developed without any particular library in consideration. It was visibly felt that many of the local functionalities required by a particular library are difficult to be customized with such generic packages. The key reasons to embark upon the indigenous software development effort include:

- **Expensive:** Let alone the initial purchase price and Annual Maintenance Contracts (AMCs) of the commercialware, modified versions of the software, platform migration and server upgrades always come with a price tag.
- **Customization:** Also the modules of the package require quite a good amount of customization at the user site for its full utilization. Even after considerable effort, the CL failed to reach the desired level of customization for all the modules.
- **False promises:** The system requirements suggested and required did not match and most of the time this lead to delays in response to queries, transactions etc.
- **Expertise:** The vendor did not give sufficient training to the local system persons as they think this may interrupt the secrecy that shrouded the package.
- **No APIs, Source code:** Asking source code to a vendor is considered as unethical. But even to change the IP address of a server without troubling your web OPAC, you should contact the vendor seems to be injustice to the client. Constant interaction with the vendor for minor modifications can be easily avoided by providing user driven programming options.
- **Not so User-friendly Interface:** Most of the time, even to achieve simple tasks, the user has to navigate back and forth a lot of screens. Added to this, is the problem of highly populated menu screens, which discards the user attention.
- **Rigidity in operations, reports etc:** Most of the time, the commercialware provides some rigid options to achieve various tasks and reports. If the user wants them to be achieved in a different way, options provided even after repeated customization were not satisfactory.
- **Security:** The biggest potential threat was the problem of server hacking by breaking the password protection. This required us to configure a mirror server for OPAC on the public network to safeguard the library server on the private network, for which the library has to purchase another expensive license.
- **Customer Support:** The customer support has been found to be not satisfactory, after a period of purchase.
- **No RDBMS:** Apart from these problems the system is lacking the presence of a RDBMS which is essential for a functional database application. It is a matter of debate about the utility of a database engine to securely hold the data in a library automation system. Those vendors who use the same support it and those do not have their own excuses about why they are doing so.

Thus the CL has decided to design and develop a robust and reliable library automation system for CL, in one way to minimize the lacunas with the existing package and on the other hand to enhance the electronic communication and work in a web enabled environment. One Asst. Librarian was pursuing M.Tech in Computer Science as part-time also contributed in a big way to this decision by setting the scene for a serious software design and development effort [3, 4].

3.1 Requirements Analysis

A library automation package has to fulfill all the housekeeping functions in a library, such as acquisition, cataloguing, serial control, circulation control, search and retrieval services, documentation, information processing and management, and office routines. As discussed earlier, since the existing work environment is a mix of computer and manual operations, each section is visited to assess the expectations of the staff for a new system and their experiences with the existing system. This stage was conducted through various means of referring the existing package, closely monitoring its use by staff, interviewing staff, browsing the different files, registers, reports, etc. generated and a work and flow study of the section. The main concern was to evolve a solution for a single integrated package, by reducing to the maximum possible or altogether eliminating the manual operations. The system flowchart of the section, with details about the input, output, and its interfacing with other sections of library, other departments of the Institute, outside firms like vendors, publishers, libraries, etc. was also arrived at. The requirements assessed for the 'Current Journal Subsystem' as a result of an involved survey as well as listening to comments of the staff concerned lead to the following points. More details regarding requirements of other sections was presented elsewhere [3, 4].

3.2 Current Journal Subsystem

In order to automate the current journal monitoring, the system has to perform the functions of:

- Inputting serial data (those data which are essential to the system)
- Ordering new serials, as and when applicable, including restarting a stopped serial
- Renewals of presently subscribed serials
- Cancellation of presently subscribed serials, if necessary.
- Accessioning of individual issues, as and when the issues are received.
- Monitoring receipt of multiple issues of journals.
- Combined issue management (sometimes issues come merged)
- Sending reminders, if necessary.
- Grace period for reminders.
- Claiming the issue, if necessary such as, request for replacement of a defective copy.
- Selective follow-up of missing issues
- Preparation of various lists like:
 - o List of periodicals received during a specified period;
 - o List of periodicals cancelled during a specified period; and
 - o List of holding with their status on shelf.
- Keeping track of the amount spent on serials subscriptions, subject or department wise, as is the case may be.
- Estimation of the budget for the next academic/financial year
- Provision for renewal alerts.
- Handling supplementary invoices and partial changes in the processed invoices.
- Accepting refunds from the suppliers and adjustment in the budget database.

3.3 Software Design

To make the design process more manageable, it can be broken down into separate steps [5].

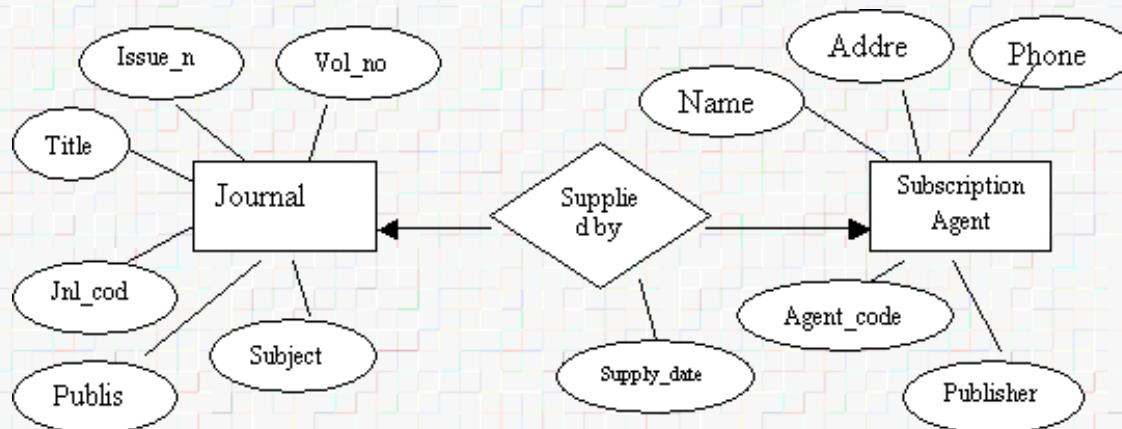
Step 1: functions or features to automate: To gain a full understanding of the real needs of the user of a new automated system and determine what functions the library would like to automate. Try to identify those repetitive and routine tasks, which now occupy a large amount of staff time.

Step 2: capabilities required: Develop a written checklist of specific functions that are essential (mandatory) and those which are desirable (optional). Be honest about what you absolutely must have vs what would be nice to have. What features are required and reports are generated by the system? Does the library should use the existing student or patron ID number of the Institute? Whether fund accounts conform to certain bookkeeping standards? How do you want clients to search the database?

Step 3: data to be stored: Decision has to be taken about the number and details of titles, volumes, patrons, orders placed, notices sent, number of vendors etc. In addition to books, materials like non-print materials, uncatalogued items, equipments, kits, software etc also have to be considered. And be sure to allow for plenty of growth; streamlined procedures can enhance usage rates significantly.

3.4 E R Diagrams

In the entity-relationship (E-R) model, the whole database can be modeled as a set of entities (distinguishable objects that exist in the real world, such as books, users, vendors, journals, binders) and relationships among them. There are certain attributes that describe the entity in the database, for an entity like book, some of the attributes are author, title, edition, call no, accn._no etc. A relationship is an association among entities like the 'supplied by' relation between vendor and book. E-R model is famous, as the graphical representation (also called E-R Diagram) helps us to structure entities, their attributes and relationships in the database design stage. In the E-R diagram, rectangles denote entities, ellipses represent attributes, diamonds stand for relations, and lines (plus arrows indicate one in relations such as one-to-one and many-to-one) link attributes to entities and relationships to entities. For clarity in understanding, each symbol is labeled with the entity or relationship it represents. The ER diagram of the system drawn helps to identify the modules and tables, normalize the different relations are to 3NF. A typical E-R diagram for the journal supply relation is shown in figure 1.



Data Flow Diagrams

Figure 1: E-R Diagram for the Journal Supply Relation

Similarly data flow diagrams (DFDs) are drawn to represent the flow of data in the system like, input data to the system, various processing carried out on the data, and the output data generated by the system. A sample DFD for the Current Journal subsystem is displayed in figure 2.

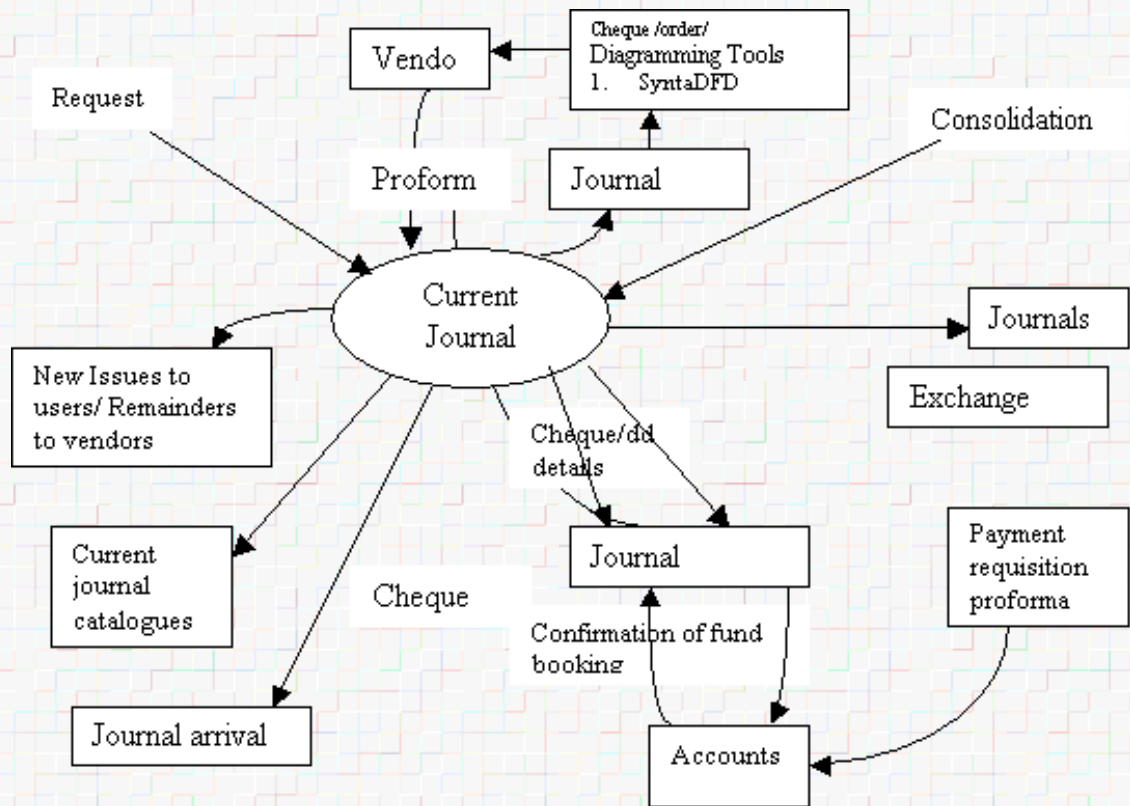


Figure 2: DFD for Current Journal Subsystem

3.5 NF Relations

Think of an unnormalised relation in which information about journals and vendors are stored in one database table. This structure is not at all feasible for storing as a file in computer as the Journal contains a lot of attributes on one hand and Supply contains different number of entries. Relations are to be normalized to avoid inconsistencies and to prevent any imminent data loss when existing relations are altered for insertion, deletion, or updation [6, 7]. Inserting new data to a relation should be possible without leaving blank

fields for some attributes. Similarly deletion of unwanted data should be effective without losing any vital information. Different relations concerning entities in the library system are being normalized to the Third Normal Form (3NF) [3].

3.6 Tables

It is difficult to discuss about all the tables prepared to deal with the different automation tasks of the Central Library, IIT Kharagpur. Some of the tables prepared for the current journal subsystem consist of `jnl_main` to collect constant information like publisher, ISSN; `jnl_arvl` table to register receipt of issues; `jnl_dept` to assign journals to different departments; `jnl_budget` to map the dept-wise budget allocation for journals; `jnl_deletn` to keep the information concerning deleted journals for back volume section and for future use by current journals in case the subscription resumes after few years; etc.

3.7 Implementation

Hardware: HP 9000 A 180 C server. Two servers are procured to keep one server on the private network of library for database operations by library staff, and another for mirrored OPAC serving on the campus Intranet and Internet.

Operating System: To have safe and secure network services on the campus Intranet, UNIX operating system is preferred over windows. Since automatic power on and booting up in the event of power breakdown and UPS expiry in the night were the vital points, HP Ux (Version 11.0) operating system is preferred over other UNIX versions.

Database Engine (backend): Oracle is selected as the database engine for its preferred virtues over other similar systems [8], and also due to its wide distribution and support in the country, and for the better support of web enabled tools like webdb.

Front-end: Oracle WebDB is used as the front-end, a complete solution for building, deploying, and proactively monitoring Web database applications and content-driven Web sites as it provides a fast and easy way to "Web-enable" Oracle databases [9].

Security: Two possibilities exist to do mirroring, one at the OS level using Unix features and another at the DBMS level with Oracle tools. Since we found the second option convenient and cost less on system resources, it is being used for real time OPAC serving, even though we successfully explored the first option.

Human Resources: The CL has been toying with the idea to fetch an R & D project sponsored by some funding agency. Whomever we approached encouraged us to either use a commercialware or to share the outcome from earlier projects executed in other Institutions. Since not convinced of these solutions, on its own funds, the CL has decided to pursue its dream of an in-house package for library automation. We recruited on temporary basis, four fresh MCA holders to implement the design using Oracle, and WebDB. They worked for more than a year to complete the implementation, data transfer from the existing commercialware, testing, trial launch in Circulation and Back Volumes sections, and in preparing documentation and documentation testing before a formal final launch of all modules.

Coding and Testing: The menus that are provided for the Current Journals Module include: TITLE APPROVAL (Consolidation, Deletion of Titles, Replacement Against Deletion, New Journals Entry, Title Approval), ORDERING (Vendor Changes by vendor/department, Place Order), PAYMENT (Invoices Entry, Contingent Details, Payment Details Entry, Bill Settlement), ISSUE MANAGEMENT (Journals Issue Entry, Claim For Missing Issues), ADMINISTRATION (Budget, Currency, Department, Publishers, Vendors), and REPORTS (Issue Reports, Budget Reports, Invoices Received, Invoices Paid, Deleted Journals, New Journals Subscribed, Current Journals Search, Vendor wise Report, Data Copying). The schematic of implementation is presented in figure 3, which shows the basic components of Oracle/WebDD used in the implementation. Menu options link to procedures or packages and if required data will be collected from forms from the user or sent to reports and data to be flashed/stored will be collected from/sent to tables, as the situation warrants. Two screenshots are displayed in figures 4 and 5. More details about implementation will be discussed

in other sources [10, 11].

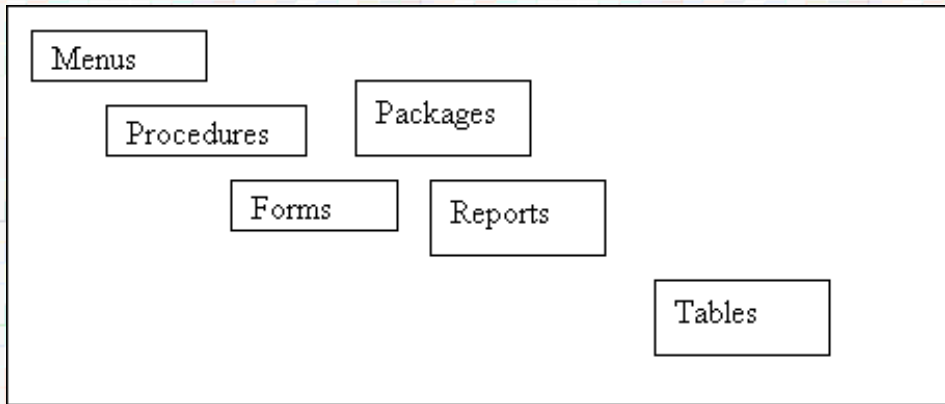


Figure 3: Schematic of Oracle tools used in Implementation

3.8 Challenges

The commercial ware is used by library staff through vt100 terminals in a purely text based mode. However, the in-house system was developed to work in a web-enabled environment. The comparatively less computer proficient staff using the package found it so inconvenient to handle mouse. This requires extensive training in mouse handling to them on the one hand and incorporating keyboard based inputs to the package on the other.

VT 100 terminals were exclusively used for commercialware applications and free from disturbances in setting. However, the PCs provided as clients to access the in-house package has wide functionalities and the staff used the same for accessing other servers as well as browsing the Internet. This sometimes leads to changes in desktop settings which they were unable to set back, and some cases even virus attacks. The design team as well as the system team of the CL had a tough time to stabilize these simple, but irritating problems.

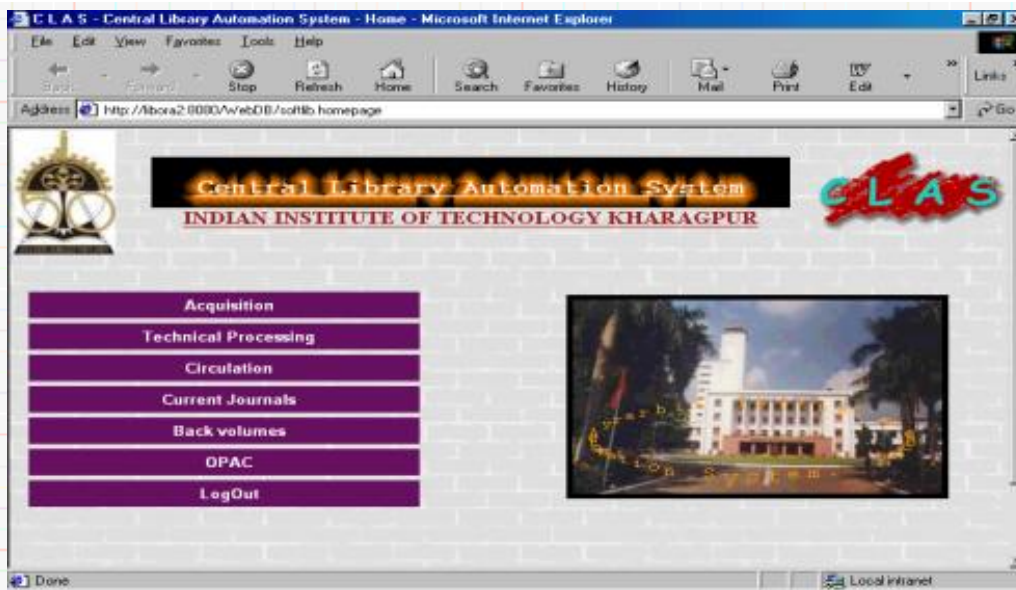


Figure 4: Welcome Screen

Private network for the existing package enables sometimes to overcome intranet problems where as the new system used the Intranet. Sometimes, these lead to delays in access and this has to be addressed from different angles. Most of the time, the design team compared the two packages and found that the terms 'slow' and 'fast' are rather relative or emotional than exact.

Some staff members who are not directly involved with the development have gone to the extent of saying that the exercise was nothing but pirating the commercialware, even though the design and development team was a genuine software development effort.

Integrating software solution always requires minor/major modifications in the procedures of individual sections to cater to the elimination of duplicating tasks and to arrive at a smooth work flow. We found that the Acquisition and Circulation section uses different codes for departments, and the two journal sections use different codes for journal. It required lot of effort from the design team to convince the individual sections about the merit of using same codes and similar procedures.

Many people see change as a threat since it often disturbs their present lifestyle. Software development projects have to address such crucial social issues to ensure the success of the project. A key ingredient for successful systems is the relationship/involvement of the user groups and system designers in the system development process [12]. As the promotions in the library are vacancy based, those not directly involved with the design and development thought that the successful implementation of the package may mar their prospects decided to put non-technical hurdles that further delayed the development work. Resolving such sensitive issues needs close monitoring by the Management and continuous counseling by the Design Team.

Figure 5: Form for updating issues of journals

Undertaking software development effort in a public Institution, and that too, in an academic institute is bound to create problems and uncertainties. The Institute and Library Authorities as well as the Design Team were open to these challenges. We failed to gather the infrastructure or competent manpower of software industry. Perhaps some of the problems faced would have been easily overcome or the whole development process would have been significantly faster with the presence of technical manpower more proficient in Oracle. Documenting a software require technical writers and this again not at the disposal of an academic institution. So the documentation face has to go through many iterations, and lead to delays in documentation testing.

Studies show that the development of a software costs only 40% (2% for requirements analysis, 12% for design, 6% for implementation, and 20% for testing) effort where as maintenance costs 60 % effort [5]. Thus the CL has to face another 60% of challenges to stabilize and maintain this package.

5. FUTURE PLANS

The prime motive behind this development has been to arrive at a very functional and state of the art package for the CL. However, the fruits of this package may be shared with other interesting libraries on a 'no profit no loss' manner. However the existing configuration of hardware and software demands at least 8 lakhs which many of libraries would find difficult to generate. So another high point of the design team is to generalize this package by reducing the system requirements, may be windows/NT based for small libraries and then for large libraries by either sourcing Oracle license at cheaper rates in bulk or to altogether use freeware/shareware like linux as the OS and database engines like MSQL. All this depends on the responses received from users and modification and testing undertaken at the CL, and the resources for this exercise gathered by the design team.

6. CONCLUSION

In-house software development has been a high point of agenda at the Central Library of IIT Kharagpur. The changing times and availability of resources and manpower the initiative with the development of a state of the art package using Oracle as backend and

WebDB as front end on HP Ux platform. The package has been developed by adopting latest procedures and trends in software engineering and by sourcing technical manpower through constant interaction with library professionals. The path traversed is not little, but the future seems to be more challenging.

ACKNOWLEDGEMENTS

The software development exercise was undertaken only due to the interest, support, encouragement and the drive to progress against all odds by the Ex-Chairman of the Central Library, Prof. R V Raja Kumar. I wish to appreciate the help rendered by M. Tech mate Gopi in the 'Requirement Analysis' stage and project fellow Rahul and ex-colleague Sainul in the trial implementation. Finally, the efforts of project fellows Bharath, Sridhar, Devendra, and Ravi helped largely to translate the design to implementation. Bharath contributed significantly in the integration and documentation of the project. I am also thankful to all colleagues of the CL for their help and suggestions.

REFERENCES

1. Jeevan, V K J, "Information Technology for Library Services at IIT, Kharagpur." *Annals for Library Science and Documentation*, 46,1, (March 1999), pp. 9-17.
2. Jeevan, V K J, "Kharagpur Electronic Library on the Internet (KELNET)" *Library Hi-Tech*, 18,3, (2000), pp. 272-278.
3. Jeevan, V K J, "Design and development of a Library Automation System using Oracle 8i," M.Tech Thesis, Indian Institute of Technology Kharagpur, 2001, 90 p.
4. Jeevan, V K J and Raja Kumar, R V, "Design of Indigenous Library Automation Software: Experiences of the Central Library, IIT Kharagpur, Information Management in e-Libraries" Proceedings of the National Conference on Information Management in e-Libraries (IMeL), New Delhi, Allied, 2002, pp. 401-410.
5. Mall, Rajib, "Fundamentals of software engineering," New Delhi, Prentice Hall, 1999, 221 p.
6. Silberschatz, Abraham, et al, "Database System Concepts" New York, McGraw-Hill, 1997, 821 p.
7. Ullman, Jeffrey D, "Principles of Database Systems" New Delhi, Galgotia, 1996, 484 p.
8. Rolland, F D. "Relational Database Management with Oracle" Boston, Addison Wesley, 1990.
9. Oracle WebDB Tutorial Guide, Release 2.2, October 1999, Part No. A77075-01, Redwood, CA, Oracle.
10. Kumar, Bharath K. et al, "Central Library Automation Software (CLAS): A Library Software, information Management in e-Libraries" Proceedings of the National Conference on Information Management in e-Libraries (IMeL), New Delhi: Allied, 2002, pp. 411-418.
11. "CLAS Documentation, Design Team" Central Library, IIT, Kharagpur, 2002
12. Coe, Larry R. "Five Small Secrets to System Success" *Information Resources Management Journal*, 9,4 (1996).

BRIEF BIOGRAPHY OF AUTHOR



V K J Jeevan is currently working as Assistant Librarian at Indian Institute of Technology Kharagpur. He was the recipient of the Young Information Scientist Award from SIS, New Delhi in 2000, the Second Prize Award for Contributing Article from RRLF, Kolkata in 2001/2 and P V Verghese Best Paper Award from ILA