

# APPLICATION OF DOMAIN SPECIFIC LANGUAGE FOR DESCRIBING AN ONLINE FILE STORAGE SYSTEM

Salla Nelson Stanley

Department of computer science, Faculty of Science Kebbi State University of Science and Technology Aliero, PMB1144 Birnin-Kebbi, Kebbi State, Nigeria.

## **ABSTRACT**

*This research Intends to develop an online File Storage/book-bank monitoring system using a well-known Domain Specific Language for Kebbi State University of Science and Technology. It was intended to address the current problems encountered in using a manual system to monitor the activities of the book-banks in Kebbi State University of Science and Technology Aliero library. Interviews, observation and record consultation were used to gather data. The research analyzes the system requirements and then came up with the requirements specifications and system specifications. The system was designed in accordance with specifications to satisfy the requirements. The system designed was implemented with MYSQL, PHP and HTML. The system is designed as an interactive and content management system which deals with data entry, validation and updating while the interactive system deals with system interaction with the users. The system is capable to largely address the problems mentioned in the existing system..*

## **KEYWORDS**

*online file storage, book bank monitoring system, domain-specific language, interactive system, content management system*

## **1. INTRODUCTION**

A File store is a structured collection of data files housed in a conventional hierarchical file system (Kathleen, 2011). Many applications use file stores as a common database, and the correct execution of these applications requires that the collection of files, directories and symbolic links stored on disk satisfy a variety of precise invariants. Moreover all of these structures must be acceptable ownership, permission and time stamp attributes.

A Domain specific language (DSL) is a programming language that is targeted towards representing problems and the solutions of a particular domain or area (Rohit, 2013). By contrast, a General programming language (GPL) is used for developing software variety of application domains Examples of commonly used DSLs are HTML, CSS, Verilog, Latex, SQL, AutoCAD, and YACC On the other hand languages such as C, Java, Perl, Python and Ruby are examples of GPLs.

Domain specific language as computer programming language of limited expressiveness focused on a particular domain. A DSL is targeted at a specific class of programming tasks; it may indeed not be clear. By restricting scope to a particular domain, one can tailor the language specifically for that domain. Common concepts or idioms in the domain can be made more easily and directly expressible even at the cost of making things outside the intended domain more difficult to write.

The assumptions common to the domain may be encoded within the language itself, so that they need not be repeated over and over for each program within the domain and again, those

assumptions may be inconsistent with applications outside the domain. The term 'DSL' is rather more recent than its meaning; DSLs have been prevalent throughout the history of computing (Fowler, 2011)

As (Mernik et al., 2005). observe, DSLs have in the past been called 'application-oriented', 'special-purpose', 'specialized', 'task-specific', and 'application' languages, and perhaps many other things too. The 'fourth-generation languages' (4GLs) popular in the 1980s were essentially DSLs for database-oriented applications, and were expected at the time to supersede general-purpose 3GLs such as Pascal and C. One might even say that Fortran and Cobol were domain-specific languages, focused on scientific and business applications respectively, although they are both not clear.

In School, reading materials are stored in libraries. The library is a place in which books and related materials are kept for use but not for sale. It is also organized for use and maintained by a public body, an institution or a private individual. In addition, it is a place in which we get information in any format and from many sources. In other to solve the problem to ensure that students accessed books at limited cost. Therefore DSL will be used to automate the higher institution library to ensure the effectiveness of the library.

## 2. ANALYSIS OF EXISTING SYSTEM

The configuration of library systems varies from one library to another, and depends on the type and complexity of the library system. The analysis of the system is divided into two areas: Analysis of the current system and the problem inherent in the existing system.

The library contains approximately 45,000 volumes of books which are open to all registered members of the university community. The staff working in the library includes:

- **Professional:** With professional qualification on library system, function and usage.
- **Para-Professional:** With lesser qualification but equipped with basic knowledge of library system, function and usage.
- **Non-Professional:** these are mainly staff without enough qualification whose job is menial i.e. shelving and stamping, drivers, cleaners, security

**The operational hours of the library are as follows:**

Period	Days	Time
During Section	Monday - Friday Saturday Sunday	8 a.m-10 p.m 8am-6pm Closed
During Vacation	Monday-Friday	8am-4pm

Table 1. Library time table

## 3. PROBLEMS OF THE EXISTING SYSTEM

The problems facing the existing systems are as follows.

- **Lack of Accuracy:** Considering the fact that human factor can set in at anytime, there are likely tendencies of mistake, especially taking statistics on library usage.
- **Poor Record:** Considering the large volume of data involved, the existing system has poor approach to large database design. The manual system makes use of a register where all registered library users have their data entered.
- **Large Manpower Requirement:** Because of the manual operation, a lot of manpower resources required.
- **Inability to Track Materials Easily:** Because of its manual operation the tracking of books is sometimes delayed.
- **End-User Statistics:** in the existing system, statistics must be taken at the end of the day's work and that is tedious and time consuming.

#### 4. REVIEW OF RELATED LITERATURE

Domain-specific language engineering reports the design and building of a WEBDSL, a DSL for web applications with a rich data model, using several DSLs for DSL engineering: SDF for syntax definition and Stratego/XT for code generation. The paper follows the stages in the development of the DSL. The contributions of the paper are three-fold. (1) A tutorial in the application of the specific SDF and Stratego/XT technology for building DSLs. (2) A description of an incremental DSL development process. (3) A domain specific language for web-applications with rich data models (Visser, 2008).

Domain-specific languages have been developed in a number of areas to facilitate the construction of models at a level closer to the conceptual model, thereby making model implementation more accessible to domain experts. Such tools to support modeling in the domain of landscape ecology can achieve a balance between the flexibility of programming and the structure and ease of using pre-built models (Andrew, 2001).

Domain Specific Language (DSL) that has been developed to enable implementation of concepts of discrete mathematics involves library of data types and functions which provides functionality which is frequently required by users. Covering the areas of Mathematical Logic, Set Theory, Functions, Graph Theory, Number Theory, Linear Algebra and Combinatorics, the language's syntax is close to the actual notation used in the specific fields (Rohit, 2013).

Domain-specific language for the formulation and implementation of atmospheric models ATMOL was developed in close collaboration with meteorologists at the Royal Netherlands Meteorological Institute KNMI to ensure ease of use, concise notation, and the adaptation of common notational conventions. ATMOL's expressiveness allows the formulation of high-level and low-level model details as language constructs for problem refinement and code synthesis. The atmospheric models specified in ATMOL are translated into efficient numerical codes with CTADEL, a tool for symbolic manipulation and code synthesis (Robert, 2001).

A website a Library Management system mainly use by member and staff allows member to search and reserve book. Member allows search the book and knows what book contain in the library. Other than that, member allows to see the book detail and also comment on the book. Member also allow reserve book via website so that they can save their time and cost to travel from their destination to library. Members also view current book they are borrowed and also history of the book they have borrowed. For the staff user, they are able to add, edit and delete the news. When new news added, the homepage news tab will be shown the new data (KOH, 2011).

Library management system developed to manage the daily book transaction and manage member, books record more efficiently. It can improve management of the book property in the library. It is mainly used by the librarian and library admin. A normal Librarian is able to manage the member maintenance module and also the most important module in the library which is book transaction module. Besides that, library management system also allows user to manage the publisher as well as lost module. On the other hand, other type of user which is admin level staff is able to handle the staff module and view report module (Tan, 2010).

Online Library Management system which maintains books in the server in any format. It allows storing books, documents, audio and video files. To use this system user must be a member in this. Then user allowed storing books, documents, A/V files. Then user can search by giving a simple query. It is difficult search books manually. Online library makes it easy (Sathish, 2012).

Library Management System software for monitoring and controlling the transactions in a library developed in java which mainly focuses on basic operations in a library like adding new member, new books, and updating new information, searching books and members and facility to borrow and return books. "Library Management System" is a windows application written for 32-bit Windows operating systems, designed to help users maintain and organize library. This software is easy to use for both beginners and advanced users (Ashutosh, 2012).

In the light of the above, updating has become automatic part of the most processes, and an update in one operation is as update for everyone, and to a very large extent, the need for replicating data on multiple pieces of paper is eliminated. Hence, automation helps the library system to operate efficiently.

## **5. METHODOLOGY**

The research work will only make use of three means of requirement elicitation which include:

### **5.1 INTERVIEW METHOD**

This technique was used because;

- It gives you accurate information on what you are researching on.
- The researcher gets final results since it is collected from the main source of information.
- The researcher is able to ask all the questions and get all the answers at that moment, since there is direct contact.
- It allows the researcher to get more information through probing and observing non-verbal messages.

### **5.2 OBSERVATION**

Observation involves taking a close look on the activities going on in a system. During our observation, we it was discovered that higher institutions make use of five method of book control, which are:

- I. Formal system of book control as the name implies involves the use of formal document like registration cards, borrower's card, date due card, author's card, order form, over-due note etc. to carry out functions that will lead to a successful book control.
- II. Informal system of book control is adopted by Kebbi State University of Science and Technology Aliero, to the formal system of book control as the latter could be not

guarantee 100% efficiency under this system, the following methods were observed by the researcher:

- A. Counting Method: This is applied when some documents are counted to take the statistics of each day's activities. For instance, the author's card and date due cards are counted to take record of the books borrowed or returned that day.
- B. Slight – Contact Method: This is a situation whereby some materials are checked thorough look on them. For instance, it helps to know books which are torn or in bad shapes, for them to be sent to the bindery for repair.
- C. Security - check Method: This involves the inspection of the library users to ensure that they do not go with books unauthorized. This is normally done at the entrance where the users are checked before they leave the library.

### 5.3 RECORD CONSULTATION

These records are library file report, library corporate policy and procedure documents, organizational chart and on line/records. And are represented using the data flow diagram as represented in the diagram below:

#### Level 0 Data Flow Diagram

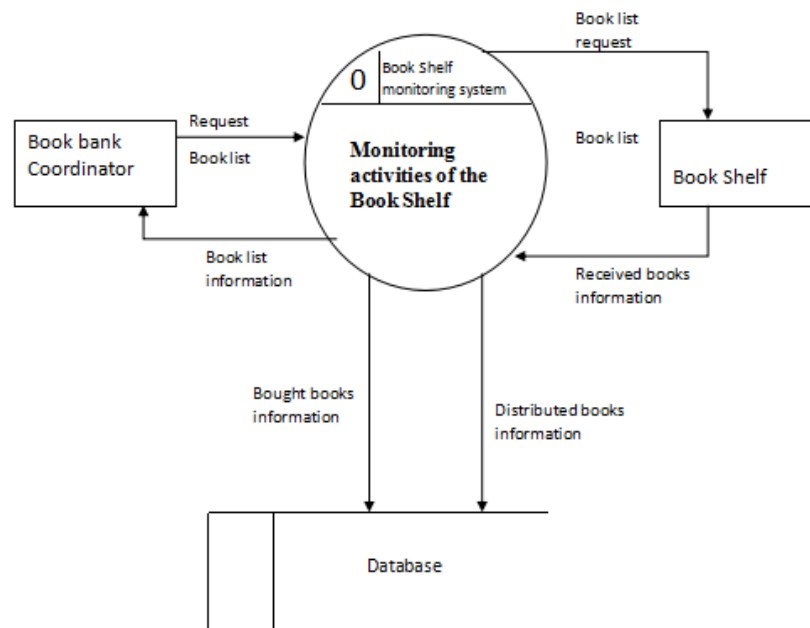


Figure 1. Architectural design

The data flow diagram (DFD) shows a functional perspective where each transformation represents a single function or process as indicated by the arrows in the figure above. The book bank coordinator is an entity that helps in processing and coordinating the activities of the library users. The book bank coordinator request information about a booklist from the book shelf monitoring system, the bookshelf monitoring system request for the booklist information from the bookshelf and the feedback sent to the monitoring activities of the book shelf. The information will be search in the database and the result sent back to the book bank coordinator.

**Context Flow Diagram of the proposed system**

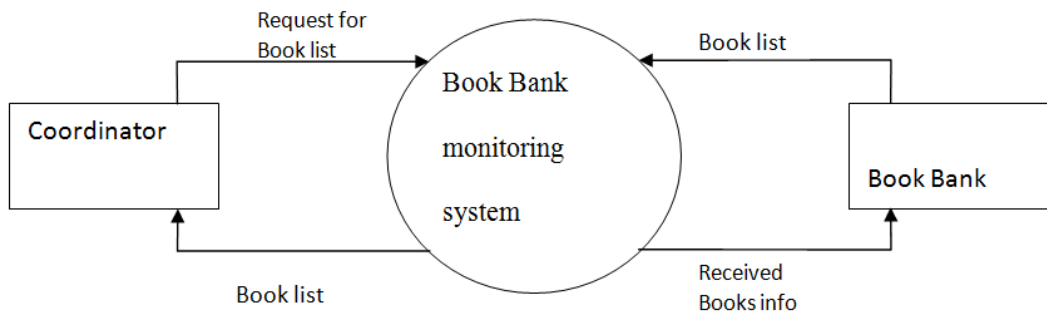


Figure 2. The context Flow Diagram of the proposed system

**Data Flow Diagram of the proposed System**

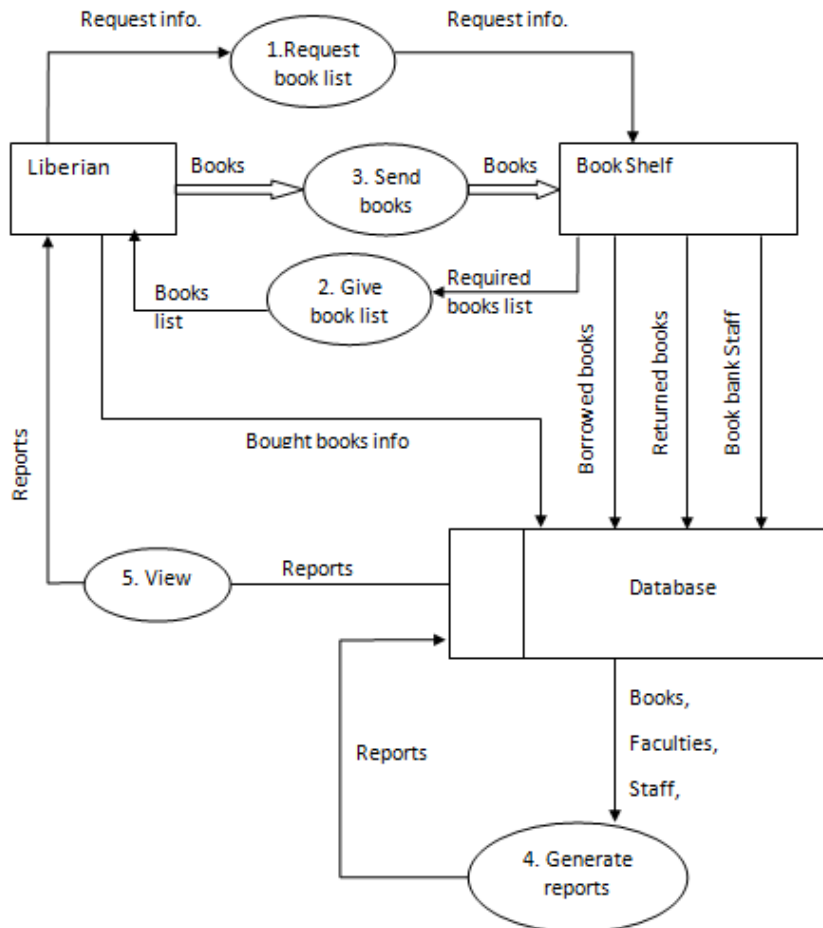


Figure 3. Level 1 Data Flow Diagram of the proposed system

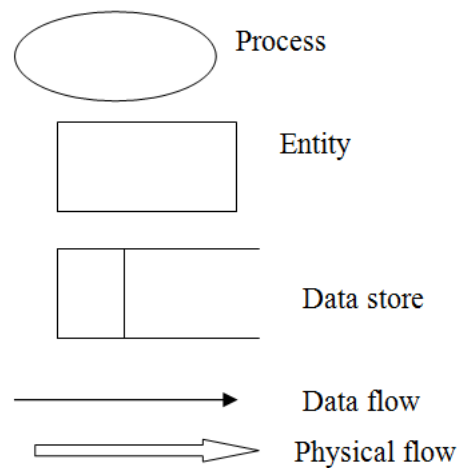


Figure 4. Key to the Data Flow Diagram

## 5.4 CONCEPTUAL DESIGN

The system has the following entities:

- Book-bank coordinator
- Book-bank staff
- Book-bank
- Student
- Lecturer
- Book
- Book copy

### 5.4.1 Entities and Attributes

- Book-bank coordinator (coordID (PK), FirstName, lastName, email)
- Book-Shelf staff (staffID (PK), FirstName, lastName, email, bookbankID (FK))
- Book-bank (book-bankID(PK), bookbankName, faculty)
- Student (regNO (PK), studentName, course, yearOfStudy)
- Lecturer (lecturerID (PK), lecturerName)
- Book (ISBN (PK), title, author, publisher, edition, borrowDate, returnDate, dueDate, NoOfCopies)
- Book copy (AccessNo(PK), title, author)

### 5.4.1 Relationships

- Coordinator Supervises Book-Bank
- BookBank staff WorksIn Book-bank
- Book has BookCopy
- Student Borrows BookCopy
- Lecturer Borrows BookCopy
- BookBank Stores Book



Figure 5. Entity Relationship Diagram of the propose system

Table 5.1. Borrower's Table

Attributes	Data Type	Description
BorrowerNo(PK)	Varchar (20)	Borrower's number
borrowerName	Varchar (30)	Borrower's first name, last name and other names.
Type	Varchar (20)	Type of a borrower whether student or lecturer
Course	Varchar (20)	Student's course if the borrower is a student.
bookTitle	Varchar (30)	Title of the book
copyNo	Varchar (20)	Accession number of a book copy
borrowDate	Date	Date when the book was borrowed
returnDate	Date	Date when the borrowed book is returned
dueDate	Date	Date when the book is due

Table 5.2. Book-bank staff

Attributes	Data Type	Description
StaffID(PK)	Varchar (10)	Uniquely identifies a book-bank staff.
firstName	Varchar (30)	The first name of the book-bank staff.
lastName	Varchar (20)	The last name of the book-bank staff.
Email	Varchar (40)	The email address of the book-bank staff
bookbankName	Varchar (20)	The name of the book-bank to which the staff belongs



Table 5.3. Book Table

Attributes	Data Type	Description
ISBN (PK)	Integer (11)	Uniquely identifies the book
Title	Varchar (40)	The title of the book
Author	Varchar (30)	The author of the book
Publisher	Varchar (40)	The publisher of the book
Edition	Integer (5)	The edition number of the book
BookbankName	Varchar (20)	The book-bank to which the book belongs
Faculty	Varchar (20)	The faculty where the book-bank is located
NoOfCopies	Integer (5)	The number of copies of a given book

Table 5.4. Book Copy Table

Attributes	Data Type	Description
AccessNo (PK)	Varchar (20)	Uniquely identifies a book copy
Title	Varchar (40)	The title of the book
Author	Varchar (30)	The author of the book
borrowDate	Date	The date when the book was borrowed
ReturnDate	Date	The date when the book returned
Borrower	Varchar (30)	The name of the borrower of a book (lecturer or student)

Table 5.5. Student Table

Attributes	Data Type	Description
RegNo (PK)	Varchar (5)	Uniquely identifies a student
firstName	Varchar (30)	The first name of a particular student
lastName	Varchar (30)	The last name of a particular student
DOB	Date	The date of birth of a student
Course	Varchar(20)	Shows the course in which the student is enrolled.
Faculty	Varchar (25)	The faculty to which the student belongs

Table 5.6. Coordinator Table

Attributes	Data Type	Description
CoordID (PK)	Varchar (10)	Uniquely identifies the bookbank
FirstName	Varchar (20)	The book-bank coordinator's first name
LastName	Varchar (20)	The book-bank coordinator's last name
Email	Varchar (40)	Book-bank coordinator's email address

Table 5.7. Lecturer Table

Attributes	Data Type	Description
lecturerID	Varchar	Uniquely identifies a particular lecturer.
lecturerName	Varchar	The name of a lecturer.

## 6.1 User Requirements for the Proposed System

The system should:

- Be able to display the list of books in the entire database to the users as per their access rights and privileges.
- Have user-friendly interfaces and user guides understandable by people of average computer skills.
- Be robust enough so that users cannot corrupt it.
- Be able to handle more than one user.

- Be maintainable by persons of average computer skills, with no knowledge about software used.
- Be able to evolve without damaging existing system.
- Have the capacity to handle expansion.
- Be friendly to both users and administrators.

## **6.2 System Requirements**

### **6.2.1 Functional Requirements**

- Storing and retrieving information on users.
- Storing and retrieving book information.
- Maintaining and manipulating records in system database through function like data entry, editing, deletion and reports.
- Create pre-designed reports on a regular basis.
- Store and retrieve records of all transactions of the various book banks.
- Validate input data
- Carry out parallel manual operations.

### **6.2.2 Functional Requirements**

- The system should be fast enough to satisfy the users.
- The system will authenticate all users by username and password before getting access to resources.
- The system should be available at any time of the day so that the library user can search for material at his/her own convenience
- The system should be accessible on the University intranet as well as any other place where there is Internet connectivity.
- The system should be maintainable by an individual with average computer capabilities. Data input, modification and elementally logical streamlining should be done using online prepared forms that will be friendly and robust enough for the user to manage with minimal possibilities of causing mistakes.
- The system shall have appropriate menus from which the user will make selections for the required functions.
- The system shall have a user-friendly interface, which can be easily learnt and used within a short time possible.
- System should be protected from abuse by unauthorized users
- The system shall be able to present output in an orderly manner.

### **6.2.3 User Specification**

The user of the system is expected to:

- Possess basic knowledge of electronic data processing
- Be familiar or able to familiarize with the terminologies used in the library.

### **6.2.4 Hardware Requirements**

The proposed system would need a computer with at least 400MB random access memory and should have a UK/US system of keyboard. In addition, the system should have a colored monitor screen with a minimum processing speed of 1 GHz, a printer preferably a laser jet 6P, mouse and pad.

### 6.2.5 Software Requirements

- The application runs under Windows 2000 or higher.
- The anti-virus software toolkit used is MacAfee anti-virus that should be regularly upgraded.
- MYSQL: Preferred because is efficient in developing database driven web sites.
- PHP: is a server side scripting language and is compatible with MYSQL.
- HTML: Used in web designing because of its being interpretable by all available web browsers and makes the directory accessible.
- Mozilla Firefox: This is a browser which is using for testing and running of program.
- Microsoft Word Processor (2007): application software used in typesetting, typing, writing letters, drawing. It was basically used in this project to type and draw tables.

## 7. IMPLEMENTATION

### 7.1 DATA PROCESSING

File Store requires a server which is used to host the on-line bookbank website and database. The server setup used in this project is known as Windows Apache MySQL PHP (WAMP). The WAMP server system consists of a number of open source software technologies which are commonly used together in server applications. It consists of an Apache server 4, MySQL 5 for managing databases, and PHP 6 scripting language used for server side data processing in dynamic web pages. These technologies are running on a Windows operating system. The WAMP system was chosen as it encapsulates the required software technologies for online book-bank application. MySQL facilitates the storage and retrieval of book from the file store system (Book-bank). This data is stored in a MySQL database. The Apache server hosts the webpages which are written in PHP. Using PHP allows interaction with the MySQL database, such as storing and retrieving data, and processes this information displaying it to the user through the website.

### 7.2 The Monitoring System

The relations were created using a script that runs on a database that was created using command prompt. Primary keys uniquely identify all entries and checks duplication while foreign keys link tables and enhance referential integrity. Data manipulation which involves; inserting, deleting, retrieving and ordering of outputs for any search was done at this level. The system was designed using Microsoft windows environment to ensure better implementations. Considering that the greatest expansion of the Internet came after the introduction of globally unique identifier to digital information (URL), the development of a Hyper Text Markup Language (HTML) and the compilation of a Hyper Text Transmission Protocol (HTTP), and forming World Wide Web in early 1990s. This means that since it is possible to access the information on the network, then HTML was used to design the web pages.

## 8. CONCLUSION

The new realities of Digital information processing require that access is provided in a widely distributed manner, and it is up to the user to locate what is needed. The ability to call up all relevant information at the click of a mouse is a researcher's dream. The Book-bank monitoring system research look at ways in which a domain specific language can be use in developing a software infrastructure that enables those who manage and maintain such collections to make them publicly available. The entire University community stands to benefit greatly from an online

book-bank monitoring system. The system will enable the update, retrieval, deletion, and generation of status reports according to the existing demands.

The newly developed online book-bank monitoring system as described in this research work solves the problems experienced using the file based manual system as it provides for quick data generation that saves time. Security is catered for through access control which requires valid users credential for both the coordinator and the book-bank staff. The system is able to generate reports, which can be used for monitoring and decision-making. The system is user friendly, as it does not require much computer skills.

## REFERENCES

- [1] Ahuan, T. C. (2010). Library Management System. Campbell University, USA.
- [2] Andrew Falla, Joseph Fall. (2003). A Domain Specific Language for Models of Landscape Dynamics. Canada: Simon Fraser University.
- [3] Ashutosh Tripathi, Ashish Srivastava. (2012). Online Library Management System. IOSR Journal of Engineering (IOSRJEN) , 2, 180-186.
- [4] Eelco, V. (2013). A case Study in Domain-Specific Language Engineering. Delft University of technology: software Engineering Research.
- [5] Englen, R. A. (2001). Domain Specific Language for Atmospheric Modeling. Florida: Florida University Press. USA.
- [6] Fowlwe, M. (2011). Domain Specific Language. Addison Wesley.
- [7] Kathleen Fisher, N. F. (2011). A Language and Toolkit For Programming with Filestores. Princeton University: Technical Report TR-904-11.
- [8] Kumar, M. S. (2012). Online Library.
- [9] Kumar, S. (2012). Online Library. New Delhi.
- [10] Mermik, M. H. (2005). When and how to develop Domain-Specific Languages. ACM Computing Surveys.
- [11] Rohit Jha, A. S. (2013). A Domain-Specific Language for Discrete Mathematics. International Journal of Computer Applications , 70,15.
- [12] YeeKeat, K. (2011). Library Management System. USA: Campbell University.