



**LIBRARY MANAGEMENT SYSTEM  
FOR  
St. JOHN'S COLLEGE  
JAFFNA**

**J.R.DANIEL NITHARSHAN**

**Registration Number: R141757**

**Index number: 1417576**

**Supervisor**

**MR. Tharmakulasingham Tharmmendra**

**BIT**

**October 2017**



**This dissertation is submitted in partial fulfillment of the requirement of the  
External Degree of Bachelor of Information Technology of the University of  
Colombo School of Computing**

## DECLARATION

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also here give consent for my dissertation, if accepted, to be made available for photocopying and any for inter library loans, and for the title, and summary to be made available to outside organizations.

Signature of candidate

*Daniel*

Date *05-11-2017*

Name of candidate: Mr. Joseph Rajaratnam Daniel Nitharshan

Counter signed by:

Signature of supervisor

*Tharmendra*

Date *05-11-2017*

Name of supervisor: Mr. Tharmakulasingham Tharmendra(BIT)

# ABSTRACT

St. John's College Jaffna was established in 1823 by the Anglican missionaries from England. St. John's College is a leading semi government school in northern Sri Lanka. The school has a library with fifteen thousand of books approximately. The library of St. John's College Jaffna (SJC) is running on manual system over the years but four years ago they had free software for library management with limited facilities. This library Management System is mainly used by Librarian and is able to manage all daily activities and members, books details. In the real world, the time is most impotent and it's going fast. So that, want to finish every activities in short period.

The proposed automated Library Management System is provide fast to complete the activities like searching for books, registering members, maintaining the Library catalogue, Update records and edit information, delete records and generating reports.

Here, it was used to develop the system through waterfall model. This includes problem definition, requirement analysis, system design, coding and Unit testing, integration and system testing, system maintenance, documentation. Requirement analysis is a stage to gather the user requirement such as functional and non-functional requirements, the system design was mainly used to design the user interface and database design. After complete the coding part, it was preceded to system testing to find the system bugs and fixed them.

The project was implemented by Visual Studio 2008 and SQL server 2008 was used as the database. Visual Studio's own reporting tool, crystal report was used for report generation. In addition, Rational Rose and Dia free and open source general-purpose diagramming software were used to draw the UML diagrams shown in this dissertation. At the end of this project a standalone application that met the client requirements was successfully implemented and installed at the client computer.

# ACKNOWLEDGEMENT

I wish to thank, the people who have helped me in making this dissertation and motivate me to provide effective software project.

Prior to all I am grateful to the University Of Colombo School Of Computing for providing the opportunity to present this project work. They contribute the standard computer studies in Sri Lanka. They initiate the BIT Degree program and formulating project work.

My sincere thanks to my supervisor Mr .T. Tharmmendra, who is guiding me and give me lot of ideas, advices to fulfill the project work. He spends his lot of valuable time to evaluate this project in his busy schedule.

I would like to thank all the Staffs of St. John's College Jaffna, Specially the Principal Rev .N.J. Gnanaponrajah who grants me the permission to undertake the software project and the Librarian & Library Assistant, who are cooperate and give all the instructions of the Library Management system of St. John's College Jaffna.

I also thank to my family and friends who have been a tower of strength, for proof reading this dissertation. They encourage me to do the project as well.

Most importantly, thanks and praise to the Jesus Christ for guiding me and for all the blessings and mercies he has bestowed on me.

# TABLE OF CONTENT

DECLARATION .....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT .....	iv
TABLE OF CONTENT .....	v
List of Figures .....	viii
List of Tables .....	ix
LIST OF ACRONYMS .....	x
Chapter 1: Introduction.....	1
1.1. Introduction Of LMS.....	1
1.2. Project Aims and Objectives .....	1
1.3. Scope of the Project.....	1
1.4. Resource Requirements.....	2
1.4.1. Hardware Requirements .....	2
1.4.2. Software Requirement.....	2
1.5. Project Schedule.....	2
1.6 System requirements .....	3
1.6.1 Hardware Requirements .....	3
1.6.2 Software Requirements .....	3
1.7. User requirements .....	3
1.8 Structure of the dissertation.....	3
1.8.1. Chapter 1 – Introduction .....	3
1.8.2 Chapter 2 – Analysis .....	4
1.8.3 Chapter 3 – Design.....	4
1.8.4 Chapter 4 Implementation .....	4
1.8.5 Chapter 5 Evaluation .....	4
1.8.6 Chapter 6 Conclusion .....	4
Chapter 2: Analysis.....	5
2.1. Project Schedule.....	5
2.2 Analyzing the existing software systems .....	5

2.3. System Objectives .....	6
2.4. Information Gathering Techniques .....	7
2.5. Software Requirement Specification.....	8
2.5.1. General Description.....	8
2.5.2. System objectives .....	9
2.5.3. System Requirements .....	9
<b>Chapter 3: Design .....</b>	<b>12</b>
3.1. Feasibility Study.....	12
3.2. Design Techniques and Phase .....	12
3.2.1. Application Structural Design .....	14
3.3. Database design.....	19
3.4. Introduction Of LMS.....	19
3.6.3 User Interface Design.....	23
<b>CHAPTER 4 : IMPLEMENTATION .....</b>	<b>26</b>
4.1Introduction .....	26
4.2 Development Tools .....	26
4.3 Code features.....	27
<b>CHAPTER 05: EVALUATION .....</b>	<b>29</b>
5.1 Requirement for a good system evaluation .....	29
5.2 Testing.....	29
5.3 Test plan.....	29
5.4 Test Case Generation .....	30
<b>CHAPTER 06: CONCLUSION .....</b>	<b>34</b>
6.1 OVERVIEW .....	34
6.2 Lessons Learnt .....	34
6.3 Future Work .....	35
<b>REFERENCES .....</b>	<b>36</b>
<b>Appendix A – System documentation .....</b>	<b>37</b>
System Manual.....	37
<b>Appendix B – Design Documentation.....</b>	<b>40</b>
<b>Appendix C – User Documentation.....</b>	<b>44</b>
<b>Appendix D – Management Reports .....</b>	<b>51</b>
<b>Appendix E - Test Results .....</b>	<b>54</b>

Appendix F – Code Listing .....58  
Appendix G - Client Certificate.....71  
GLOSSARY .....72

# List of Figures

<i>Figure 1.1: Gantt chart for the LMS</i> .....	2
<i>Figure 3.1: Use Case Diagram</i> .....	14
<i>Figure 3.2: Borrow books use case</i> .....	16
<i>Figure 3.3: Reserve Book Use Case</i> .....	18
<i>Figure 3. 4: Return Book Use Case</i> .....	21
<i>Figure 3.5: ER diagram</i> .....	22
<i>Figure 3.6: Main Screen of the system</i> .....	24
<i>Figure 3.7: Confirm Update, Delete Dialogs</i> .....	24
<i>Figure 3.8: Add Book Form</i> .....	25
<i>Figure 5.1: Member registration Test Result</i> .....	31
<i>Figure 5.2: Book Details Test Result</i> .....	32
<i>Figure A.1: Install .NET Framework</i> .....	37
<i>Figure A.2: Install SQL Server 2008</i> .....	38
<i>Figure A.3: Create New Database</i> .....	38
<i>Figure A.4: Database Created</i> .....	39
<i>Figure A.5: Install Application</i> .....	39
<i>Figure B.1: Activity Diagram for Member Registration</i> .....	40
<i>Figure B.2: Activity Diagram for Book Returning</i> .....	41
<i>Figure B.3: Activity Diagram for Lending Books</i> .....	42
<i>Figure B.4: Sequence Diagram for Lending a Book</i> .....	43
<i>Figure B.5: Sequence Diagram for returning a book</i> .....	43
<i>Figure C.1: Login Screen</i> .....	44
<i>Figure C.2: Main Screen</i> .....	44
<i>Figure C.3: Main Tasks</i> .....	46
<i>Figure C.4: Menu Items</i> .....	46
<i>Figure C.5: Report Submenu Items</i> .....	46
<i>Figure C.6: Edit Submenu Items</i> .....	47
<i>Figure C.7: Register Members Form</i> .....	47
<i>Figure C.8: Create new user account form</i> .....	48
<i>Figure C.9: Change Password Form</i> .....	48
<i>Figure C.10: Delete User Account Form</i> .....	49
<i>Figure C.11: Fine Form</i> .....	49
<i>Figure C.12: Return Form</i> .....	50
<i>Figure D.1: New Books Acquisition Report</i> .....	51
<i>Figure D.2: Members Report</i> .....	51
<i>Figure D.3: Furniture Details Report Details</i> .....	52
<i>Figure D.4: Author Report</i> .....	52
<i>Figure D.5: Magazine Report</i> .....	53
<i>Figure D.6: Lending Report</i> .....	53



# List of Tables

<i>Table 3.1: Use Case Narrative for Registration .....</i>	<i>15</i>
<i>Table 3.2: Use Case Narrative for Lending Books .....</i>	<i>17</i>
<i>Table 3.3: Use Case Narrative for Lending Books .....</i>	<i>18</i>
<i>Table 3.4: Use Case Narrative for Returning Books .....</i>	<i>21</i>
<i>Table 5.1: Register Members Test Results.....</i>	<i>30</i>
<i>Table 5.2: Add Books Test Results .....</i>	<i>31</i>
<i>Table 5.3: Delete Books Test Results.....</i>	<i>32</i>
<i>Table 5.4: Delete Magazines Test Results .....</i>	<i>33</i>
<i>Table 5.5: Lending Books Test Results .....</i>	<i>33</i>
<i>Table E.1: Add Magazines Test Results .....</i>	<i>54</i>
<i>Table E.2: Search Books Test Results .....</i>	<i>54</i>
<i>Table E.3: Search Magazines Test Results .....</i>	<i>55</i>
<i>Table E.4: Search Members Test Results.....</i>	<i>55</i>
<i>Table E.5: Edit Members Test Results .....</i>	<i>55</i>
<i>Table E.6: Return Books Test Results .....</i>	<i>56</i>
<i>Table E.7: Lending Test Results.....</i>	<i>56</i>

# **LIST OF ACRONYMS**

GB – Giga Byte

GHz – Giga Hertz

GUI – Graphical User Interface

IDE – Integrated Development Language

IT – Information Technology

LMS – Library Management System

RAM – Random Access Memory

SJC – St. John's College

SQL – Structured Query Language

UCSC– University of Colombo School of Computing

UML – Unified Modeling Language

# Chapter 1: Introduction

## 1.1. Introduction Of LMS

The Library Management System was developed and designed to help the librarian to keep the records about the books, book lending, fine or penalties so that the problem such as file missing or record Missing will not happened again.

Book and member maintenance module also included in Library Management System. The Library Management System for **St. John's College Jaffna** mostly a Library Control, Lend and Return Management and accounting section was not included In Library Management System it was maintained by SJC library. It can also be well thought-out as a Management Information System. It will be added efficient and protected system than the existing manual system.

## 1.2. Project Aims and Objectives

Proposed system is an automated Library Management System. Through this software librarian can:

- Maintaining the library catalogue.
- Search books and members
- Add and delete books
- Add and delete members
- Update records and edit information
- Generating reports
- Maintaining the Accession Register
- Managing the lending services

## 1.3. Scope of the Project

The main scope of the project is to offer a capable and operational system as a Library Management System for the St. John's College, Jaffna to handle their daily routine activities such as user account maintenance, member details maintenance, book maintenance, lending, reservation, proper alert facilities and generating management reports for future decision making. The proposed system will facilitate the librarian in the conduct of his or her routine activities easily.

## 1.4. Resource Requirements

### 1.4.1. Hardware Requirements

#### A personal Computer With

- Pentium IV or more with 2.0 GHz Processor or above
- 2GB of RAM
- Minimum 40GB Hard disk or Above
- Dot matrix printer

### 1.4.2. Software Requirement

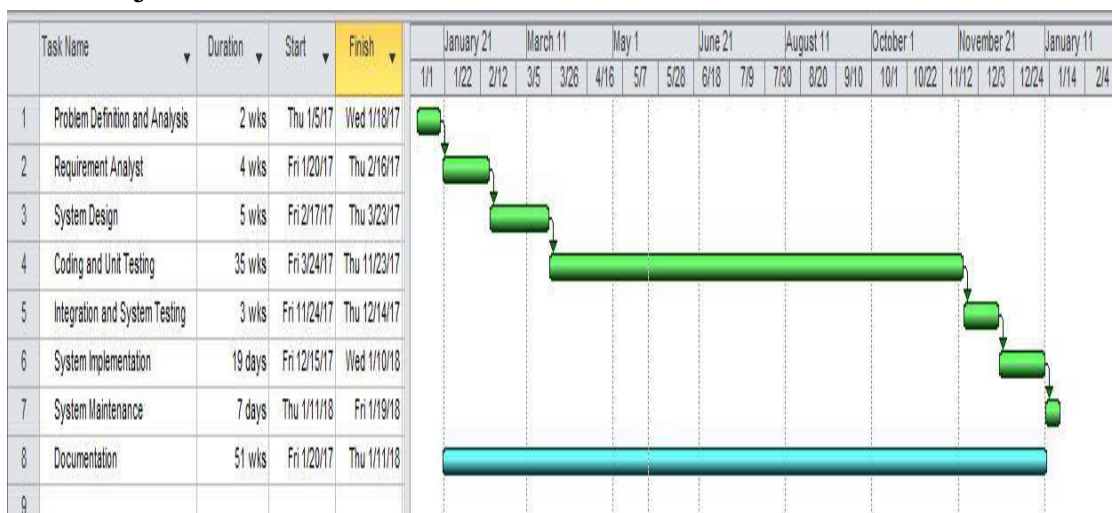
#### For Development

- Microsoft Visual Studio .Net 2008 or above
- Microsoft SQL server 2008 or above
- Crystal Report (This is available in Visual Studio .NET)

#### For Implementation

- Windows 7/ 8/10
- Microsoft SQL server 2008
- .NET framework 3.5 or above
- Crystal report

## 1.5. Project Schedule



*Figure 1.1: Gantt chart for the LMS*

The figure 1.1 illustrates the schedule of the project.

## 1.6 System requirements

### 1.6.1 Hardware Requirements

- Pentium IV or more with 2GHz Processor or above
- 1 GB RAM or Above (2GB RAM recommended)
- Minimum 40GB Hard disk or above.
- Printer : Dot matrix or Laser

### 1.6.2 Software Requirements

- Windows 7 or Higher version
- Microsoft Visual Basic .Net 2008
- Microsoft SQL Server 2008
- Microsoft .NET Frame Work Version 3.5
- Crystal Report (This is available in .Net)

## 1.7. User requirements

- Basic knowledge to handle Microsoft Windows 7
- Knowledge to operate Printers

## 1.8 Structure of the dissertation

This dissertation is written in six Chapters with main captions and sub-headings are given where more clarifications need. It describes the path and methods used to carry out the activities of each stage of the project. The outcome of the work done are then used to write the project with all the details necessary for the proposed system giving explanations, appropriate diagrams as well as graphs and reports under each chapter to the best of their relevance. The six chapters are as follows:

### 1.8.1. Chapter 1 – Introduction

The subject describes the introduction of the new system to St. John's College Jaffna , instead of the current manual system practiced. It enumerates the advantages of the system with the support of the need in the background. The target to be achieved through implementation is also defined.

### 1.8.2 Chapter 2 – Analysis

The application of fundamental requirements for this project is stated here. The analysis of the present manual system and the defects are gathered so that to include them to rectify in the proposed system. In addition, it gives the details of data collection needed and briefly the method used to carry out the project.

### 1.8.3 Chapter 3 – Design

This chapter defines the tools and techniques used to plan the method on the basis of analysis made. The system design is discussed reasoning for such decisions and also includes Database and User interface designs. Application of additional object oriented factors that were discussed during the study of the course is included as a complement of the project in a structured manner.

### 1.8.4 Chapter 4 Implementation

This fourth chapter will cover all the issues in implementation. Here the important codes are listed and explained. Implementation environment such as hardware and software and other related issues are summarized.

### 1.8.5 Chapter 5 Evaluation

This fifth chapter based on testing. It provides testing approaches and methods which were used in the project with the entire test plans, test cases and test result

### 1.8.6 Chapter 6 Conclusion

Sixth and final chapter will conclude the dissertation with critical evaluation of the system and suggestions for any future work.

# Chapter 2: Analysis

## 2.1. Project Schedule

There are two main activities in this phase: understanding or analysis and problem specification requirements. In the analysis of problems, the objective is to understand the problem and its context, and the requirements of the new system to be developed. Once the problem is analyzed and understood the essential requirements must be specified in the requirements specification document. Specifications shall specify all functional and performance requirements; the input and output format, etc.

In this chapter, we will discuss and analysis on the process of development of the library Management system including software requirement specification and Data Flow Diagram (DFD). The functional and non-functional requirement is included in software requirement specification party to provide a comprehensive overview and description of system requirements before the development process is made. In addition, DFD provides a view of how the system or company that streams able to increase effectiveness in achieving the objectives of the system.

## 2.2 Analyzing the existing software systems

Koha is a web-based Integrated Library System, with a SQL database (MySQL preferred) backend with cataloguing data stored [1]. The user interface is very configurable and adaptable and has been translated into many languages. Koha has most of the features that would be expected in an Integrated Library System, including:

- Various Web 2.0 facilities like tagging, comment, Social sharing
- Union catalog facility
- Customizable search
- Circulation and borrower management
- Full acquisitions system including budgets and pricing information (including supplier and currency conversion)
- Simple acquisitions system for the smaller library

- Ability to cope with any number of branches, patrons, patron categories, item categories, items, currencies and other data
- Serials system for magazines or newspapers
- Reporting
- Reading lists for members
- On- line Circulation
- Segmentation of the line

Evergreen is an open source Integrated Library System (ILS), initially developed by the Georgia Public Library Service for Public Information Network for Electronic Services (PINES), a statewide resource-sharing consortium with over 270 member libraries. Development priorities for Evergreen are that it be stable, robust, flexible, secure, and user-friendly [2].

Though, they are mainly based for online process and needs training to use and customize. But, the proposed LMS for St. Johns College, Jaffna has customized features with user friendly interfaces

### 2.3. System Objectives

Improved control and performance of the system has been developed to solve the current problem occurred in the library. The system must be able to validate the user, registration and memory bug.

Except cost implementation of LMS computerized library can hire 1 or 2 staff members to manage the transaction process book. With the aid of computer system, a library can save the cost of an employee leasing and also save paper costs.

Save the time of librarian and is able to locate the folder in a short time by pressing only a few keys. Compared to previous time before the implementation of the system, librarian can save a lot of time.



## 2.4. Information Gathering Techniques

Various techniques are available. Technical collection requirements can vary from one project to another. Some techniques requirements collection can be very beneficial for you in a single project, but cannot be as useful in another project or for another system. Therefore, the usefulness of art is determined by the need and type of benefits to a particular project [3].

There are some ways to discover the information in a system development project. Such as:

- Interviewing
- Observation.
- Questionnaires
- Prototyping
- Sampling

An analyst can usually apply more than one of these techniques in a project to develop a single system. In this project, it was decided that the interview and observation are appropriate to the client's environment are successful

### **Interviewing**

Interviews with user it's important in creating wonderful software. Without knowing the expectations and goals of user, we should also understand the point of view of all interviewees to respond correctly and weigh their inputs. Like a good analyst, listening is a quality that helps a great analyst to gain a better value through an interview compared to a mean analyst.

### **Observation**

Observation is an ability which the analysts have to implement. The analysts have to observe the right information and decide the right person and look at the right place to reach his aim. He should have a clear vision of how each site work and work flow between them and for this is should be a good observer.

## 2.5. Software Requirement Specification

### 2.5.1. General Description

- **Product Description**

Library Management System is an automated system which can help librarian to manage the library daily activity in electronic format. It reduces the risk of paper work such as file lost, file damage and time-consuming. Problem Statement The problem occurred before having computerized system includes:

- **File missing**

When no computerized system is implemented, the file always lost because of human and environment. Sometimes librarian didn't keep the record to its original place because of a lot member queue up to borrow books. After that the file was missing due to messy environment.

- **File damaged**

In the other possibility, the file/record will be damaged due to accident. For example the librarian accidentally hit a glass of drink and pours onto the paper file. The record will be damaged. Besides this, natural disaster such as flood also will cause damage to the file record.

- **Not easy to find record**

Without computerized system, when member wants to borrow a book, librarian hard to search for the member's record. It will cause time-consuming when a lot member are waiting to borrow the books.

- **Space-consuming**

After long operation time of the library, the records are getting more and more. Finally, the physical record was space-consuming and no place to keep the file.

- **Not easy to view reports**

Report need to generate manually without computerized system. Admin need to get the book transaction record and find the information based on the time period. It is time consuming to generate one report.

## 2.5.2. System objectives

- **Improvement in control and performance**

The system was developed to eliminate the current problem occurred in library. The system must be able to validate the user, store the record and error free.

- **Save cost**

After implementing the computerized library system, library can save the cost of hire employee and also save the paper-cost.

- **Save time**

Librarian is able to search the record in short time by pressing only few keys. to previous time before implementing the system, librarian can save a lot of time

## 2.5.3. System Requirements

### **Non-functional Requirements**

- **Products Requirements**

Efficiency requirements with the LMS, user (librarian) should be able to course of action faster when they process book deal.

- **Reliability requirements**

The system must perform accurately to request membership. For example, when the librarian records details of modified profile after they review their details, the details must be change according to the latest information they have updates. When member returns the book after the date has expired, the fine should be calculated accurately.

- **Usability requirements**

This system must be designed with user-friendly by the staff so that the user can perform their job well. It must have a clear instruction to guide the user in the system. This is also the description of the error messages should be clear.

- **Organizational Requirements**

Requirements implemented in the implementation of the system, it uses the language vb.net and key programming tools. In addition, the SQL language is used to maintain information in the database. On the other hand, SQL Server 2008 must be installed.

- **Delivery requirements**

The whole system is estimated to be about particular time and documentation will be done in short time. The complete system will be provided in softcopy documentation process while the book is in print and electronic version.

- **External Requirements**

Legislative requirements using the information must be recognized by the authorized people so he did not break the law. This information is protected by copyright and law. In addition, when visitors to become a member of the library, he or she must agree with the rules in the system.

- **Security requirements**

The system must be highly secured in the connecting portion. This is because the report cannot see by level of government. The staff can perform most of the processes, except the display module and reporting module log file.

## **Functional Requirements**

### **User Account**

- The system must be able to carry out authorization process which decides what the user's level can access to.
- The system only allows the user with valid id and password to enter the system.
- The user must able to logout after they finished using the system.

### **Add New Book**

- The system must be able to search the database based on selected search type (bookId, book title, etc).
- The System must able to find the book based on the keyword entered.
- The system must be find book details with bookId , edit the book details and update.

### **Publisher Maintenance**

- The system must be able to auto generate the publisherId.
- The system must not generate the same publisherId.
- When user clicks the find button, system should show out the publisher information.
- The system must be able to find book details with publisherId , edit the publisher details and update.

### **Member Maintenance**

- The system must be able to auto generate the memberId.
- The system must not generate the same memberId.
- When user clicks the find button, system should show out the member information.

### **Generate reports**

- System must be able to confirm away the correct information about the transaction happens on particular date.
- The same book should not appear repeatedly.
- The system must be allowed to show the book details in particular classId.

### **Activity Log**

- System must be able to show out all the activity log information

# Chapter 3: Design

This chapter will discuss about System Design phase which is one of the System Development Life Cycle phase. The Application architecture design, GUI design, database design will be carried out in this chapter.

## 3.1. Feasibility Study

During this phase you create the detailed specification for the proposed software. The objects found during the analysis phase are refined, and the database is modeled. UML diagrams such as class diagrams, activity diagrams and sequence diagrams are used for this purpose. The user interfaces are designed.

The design phase uses the information gathered previously in order to realize the logical design of the information system. It is the design of user interfaces, database and outputs of the design with users to meet their information needs [4].

Design of the information system focused on the technical implementation of the project or system development. Analysis phase of system development project is the implementation of the system independently.

Designer converts the system requirements analysis phase requirements into technical solutions. System design considers the software architecture, database design and interface design.

## 3.2. Design Techniques and Phase

The information gathered earlier in order to realize the logical design of the information system is used in the design phase of the system. Object-oriented design technique was selected after comparing to other design strategies in order to deal easily design problems like; reusability, modularity, abstraction, encapsulation and support maintainability, etc. . .

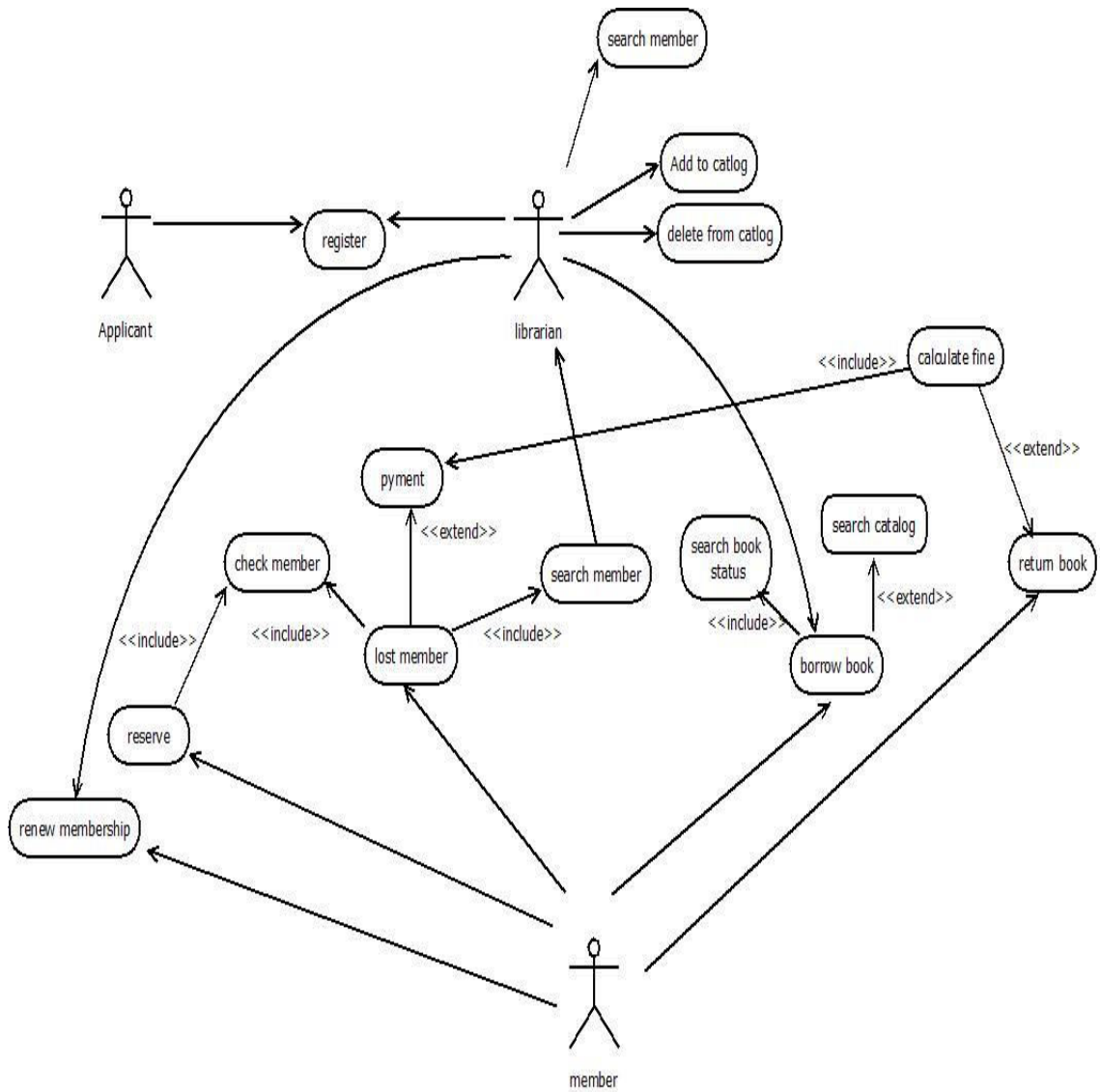
This technique is followed in the development of the Library System Management Jaffna College Institute of Technology. UML is used to provide a range of different use cases that are used to document the object-oriented design. Data Layer is designed with the help

of the identified attributes of classes. More details on the structure of the database are presented below in Database Design section in this chapter.

Design phase of the project's main activity is the development of the system. The results of the analysis phase of the system used as input for the design phase. The project was completed in two phases and are

- Application architecture design
- Data base design
- GUI design

### 3.2.1. Application Structural Design



**Figure 3.1: Use Case Diagram**

The above, figure 3.1 is the Top Level Use Case diagram for the proposed system.



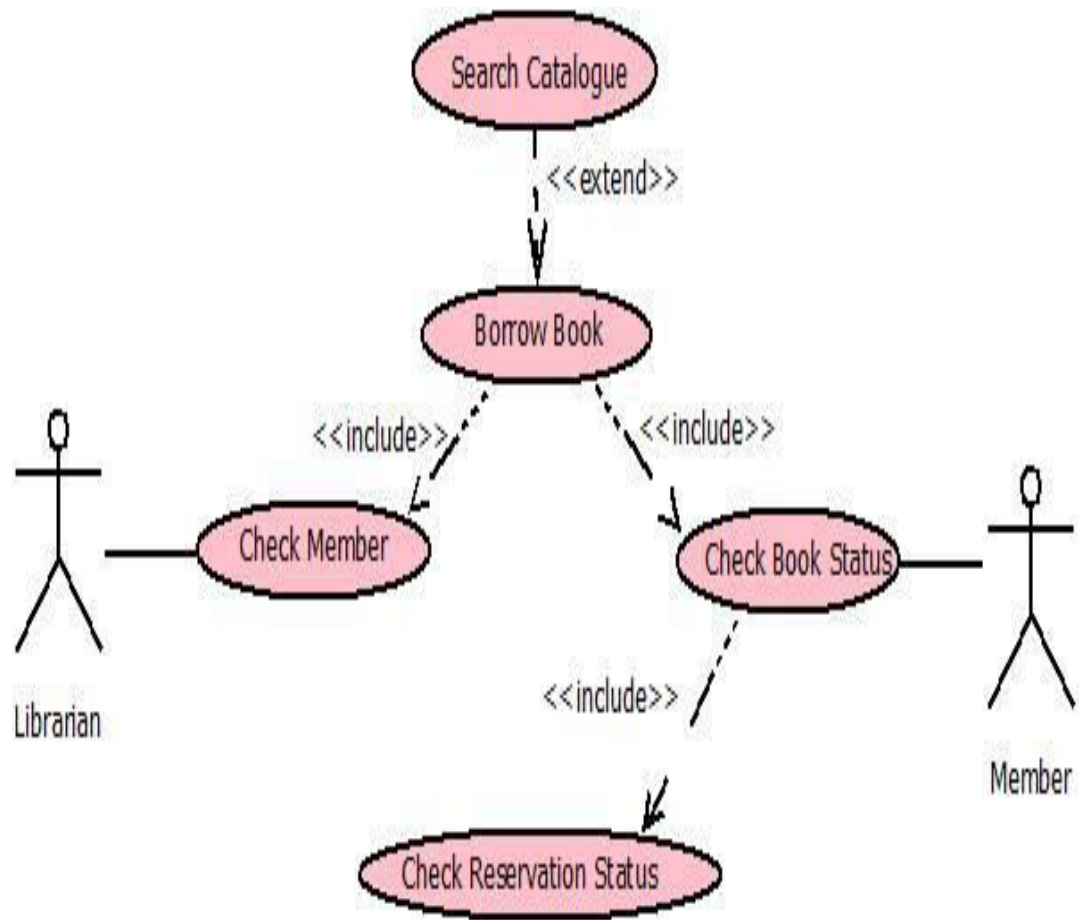
## Members Registration Use Case

Use Case Name:	Registration of Members	
Actors:	1) Applicant 2) Librarian	
Description:	Use Case describes the event of a student or staff member of St. John's College Jaffna registering for Library membership. The applicant fills out the relevant registration form, which is checked by the Librarian. If the data entered is accurate, the membership is created.	
Precondition:	Applicant is not already registered as a member.	
Typical Course of Events	<p>Actor Action</p> <p>Step 1 : The applicant fills form and provides necessary data to the librarian</p> <p>Step 2: Librarian checks data, and enters it into the system.</p> <p>Step 3: System verifies that all information is in the correct format.</p>	<p>Step 4: System checks if such a member already exists.</p> <p>Step5: Creates Membership</p> <p>Step 6 : Writes Membership cards and hands them over to the librarian and Displays member information to the librarian</p>
Alternative Courses :	<p>Alt Step 2: All information is not provided, Librarian asks the Applicant to re fill the form.</p> <p>Alt Step 3: All information is not in the correct format; system prompts the librarian to re submit the applicant information.</p> <p>Alt Step 4: System stops processing the registration request and informs the librarian that such a member already exists in the database.</p>	
Conclusion :	The use case concludes when the library cards are handed over to the new member.	
Post Conditions :	New member is recorded and registration start, end dates are set to the current registration period.	
Assumptions:	Student applicant has paid school fees.	

***Table 3.1: Use Case Narrative for Registration***

The above, table 3.1 describes the member registration Use Case for the proposed system.

## Lend books



*Figure 3.2: Borrow books use case*

The above, figure 3.2 describes the lending Use Case for the proposed system.

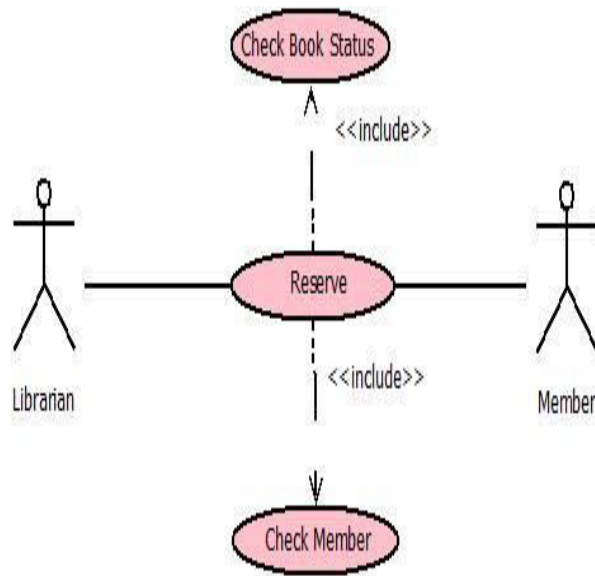
## Use Case Book Lending

Use Case Name:	Lend Books	
Actors:	1) Member 2) Librarian	
Description:	Use Case describes the event of a member Lending the book from the library.	
Precondition:	Member registration is current.	
Typical Course of Events	<p>Actor Action</p> <p>Step 1: Member searches the library manually, selects a book and provides the book and the membership cards to the librarian.</p> <p>Step 2: Librarian enters the book Acquisition Number and Membership No into the system.</p> <p>Step 9: Librarian marks the return date in the book and hands it over to the member.</p>	<p>System Response</p> <p>Step 3: System verifies the member and the book exists in the library.</p> <p>Step 4: System checks if the book condition is good.</p> <p>Step5: System check if the member has reached a Lending limit of one book.</p> <p>Step 6: System checks if the book is reserved</p> <p>Step 7: System checks if the return date of the system coincides with a holiday that is longer than 7 days.</p> <p>Step 8: System, calculates the return date and shows the information to the librarian.</p>
Alternative Courses :	Alt Step 3, 4, 5, 6, 7: System stops processing the Lending request and informs the librarian that these conditions are not met.	
Conclusion :	The use case concludes the member receives the book.	
Post Conditions :	The Lending information is recorded in the database.	
Assumptions:	Member has not outstanding fines.	

***Table 3.2: Use Case Narrative for Lending Books***

The above, table 3.2 describes the lending Use Case for the proposed system.

## Reserve Book



**Figure 3.3: Reserve Book Use Case**

The above, figure 3.3 describes the reserve Use Case for the proposed system.

## Reserve Book

Use Case Name:	Reserving Books	
Actors:	1) Member 2) Librarian	
Description:	Use Case describes the event of a staff member reserving a book from the library.	
Precondition:	Book is unavailable.	
Typical Course of Events	Actor Action Step 1: Member checks the availability of a book from the system with the aid of the librarian. Step 3: Librarian picks the copy and reserves it.	System Response Step 2: System displays the borrowed copies of the book to the librarian.
Conclusion :	The use case concludes when the book is reserved.	
Post Conditions :	The reservation information is recorded in the database.	
Assumptions:	Member is a staff member.	

**Table 3.3: Use Case Narrative for Lending Books**

The above, table 3.3 describes the reserve Use Case for the proposed system.

### 3.3. Database design

The process of database design includes the detailed design of the database based on the ER diagram created in the analysis phase. Data models, which were developed during the modeling phase of the data, are translated into data structures suitable for the selected database technology.

Data model of the SCS system was defined in the requirements specification through ER diagrams. An ER model is a conceptual representation of real objects. The main work of database design is mapping ER models to relational models. The ER diagram is transformed into a relational model. Normalization rules have been applied to relational schema. But, my ER diagram is in normalized form because of user identification and system performance [5].

### 3.4. Introduction Of LMS

#### **Normalization**

Normalization is a process of decomposing relations unacceptable in smaller ratios. The purpose of normalization is to maintain the consistency, no redundancy and reduce the space of the reports data storage. There are several levels of normalization.

#### **First Normal Form (1NF)**

Eliminate duplicate columns from the same table. Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

#### **Second Normal Form (2NF)**

Meet all the requirements of the first normal form. Remove subsets of data that apply to multiple rows of a table and place them in separate tables. Create relationships between these new tables and their predecessors through the use of foreign keys.

#### **Third Normal Form (3NF)**

Meet all the requirements of the second normal form. Remove columns that are not dependent on the primary key.

### **Fourth Normal Form (4NF)**

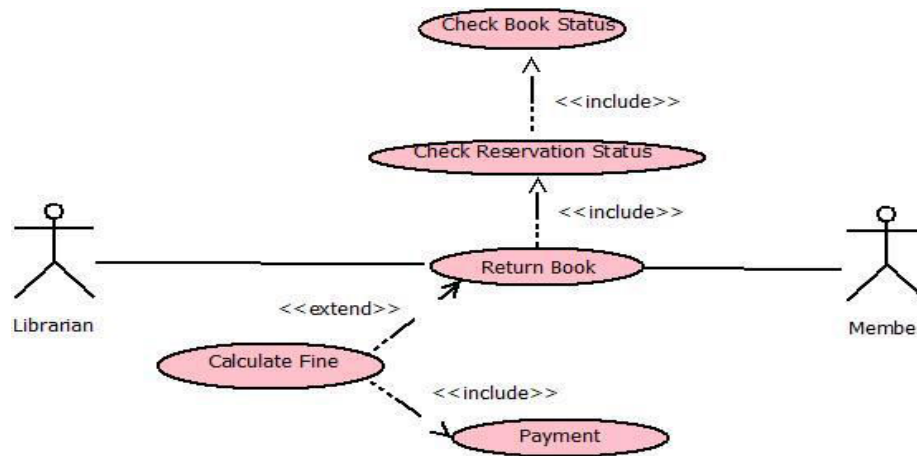
Meet all the requirements of the third normal form. A relation is in 4NF if it has no multi-valued dependencies.

The Library Management System database identified attributes of the following classes.

- Books
- Author
- Category
- Publisher
- News Paper
- Magazine
- Members
- Staffs
- Students
- Landings
- Lending Info
- Fines

Database tables are designed and they are normalized up to third normal form. Attributes identified by drawing the class diagrams were used as fields in the database tables. However since data anomalies that lead to data redundancy and loss of data integrity may still occur, the database was normalized up to 3rd Normal Form.

## Return Book



**Figure 3. 4: Return Book Use Case**

The above, figure 3.4 describes the book return Use Case for the proposed system.

## Use Case Book Return

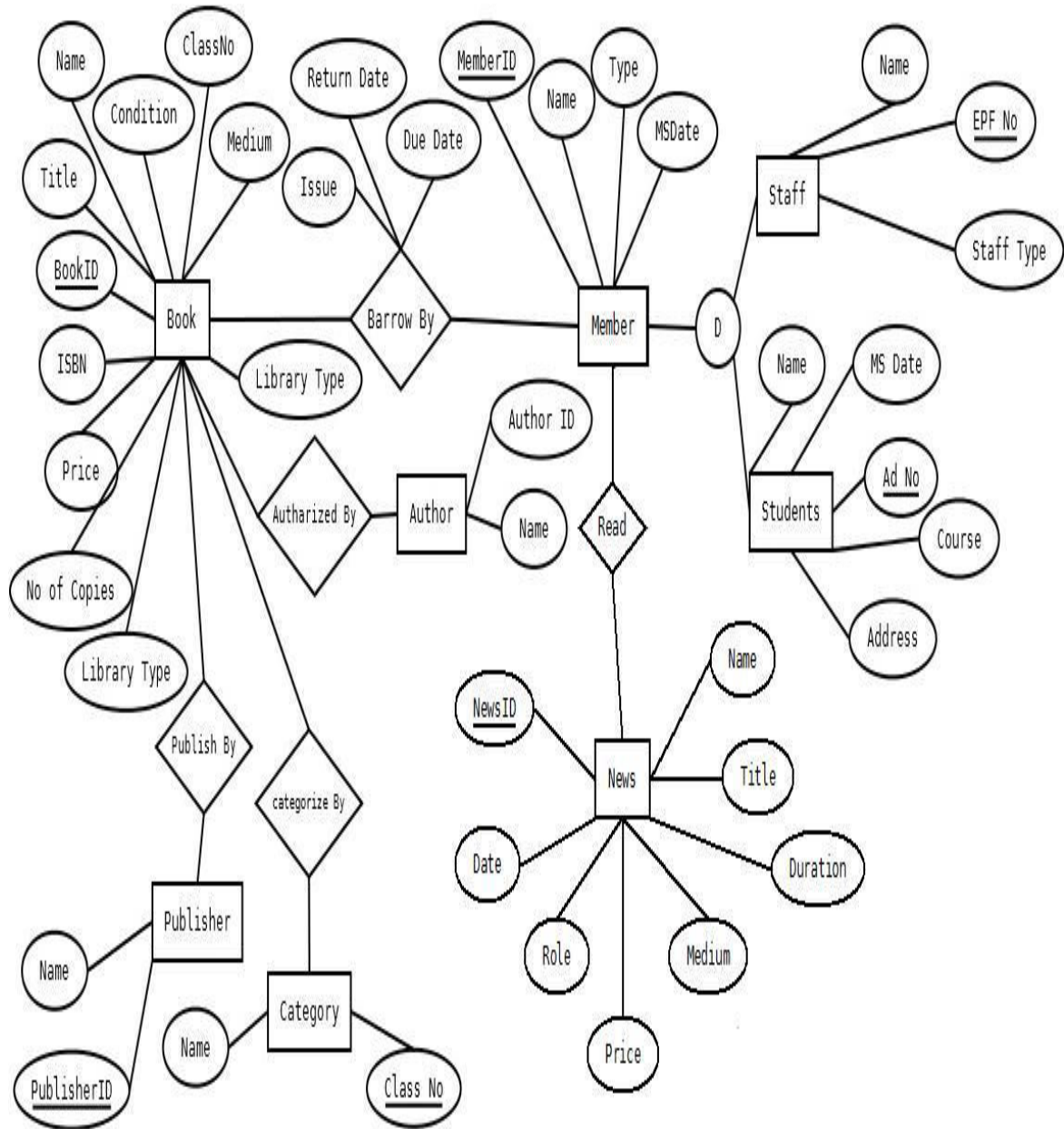
Use Case Name:	Returning Books	
Actors:	1) Member 2) Librarian	
Description:	Use Case describes the event of a member returning a borrowed book. .	
Precondition:	The book has not been returned.	
Typical Course of Events	Actor Action Step 1: Member hands the borrowed book to the librarian. Step 2: The librarian enters the book Acquisition Number into the system.	System Response Step 3: System checks if the book carries a fine. Shows book, member and borrowing information to the librarian. Step 4: System marks book as available in the database.
Alternative Courses	Alt Step 3: If there is a fine, the member pays the fine and the system, records the payment.	
Conclusion :	The use case concludes when the book is placed in the library and all fines are paid.	
Post Conditions :	If reservations exist, they are updated and the librarian is notified.	

**Table 3.4: Use Case Narrative for Returning Books**

The above, table 3.4 describes the book return Use Case for the proposed system.

### 3.5. ER Diagram

Basic ER diagram view of LMS [6].



**Figure 3.5: ER diagram**

The above, figure 3.5 describes the basic ER model of proposed system.



### 3.6.3 User Interface Design

Since the proposed system is used primarily by the Librarian of St. John's College Jaffna, the user interface must be designed so that a person of basic to intermediate computer skills will be able to interact with the proposed system with the least amount of training.

Some of the properties a user interface must have to achieve this are as follows:

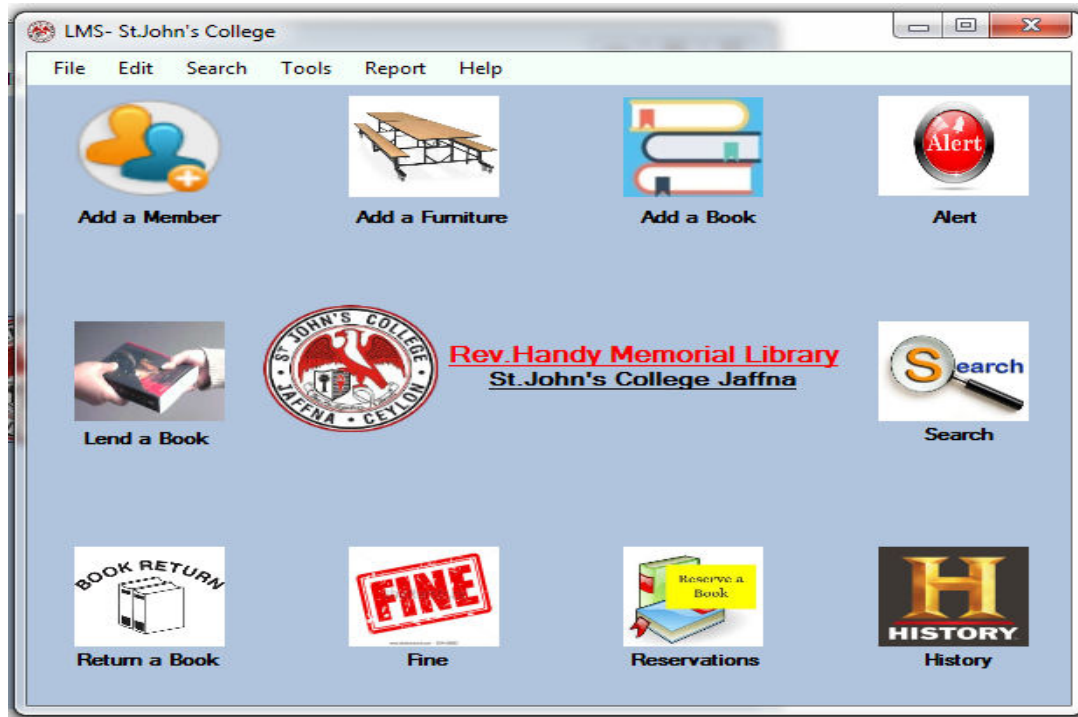
- Consistent – icons, controls, should be consistent across screens
- Simple – making complex tasks feel simple and easy to accomplish
- Provide Feedback – e.g. Error messages, confirmation messages
- Be descriptive – words used in controls must be descriptive of the task it performs.
- Easy navigation - navigation between screens must be done in a meaningful manner
- Visibility – Controls required for a task should be clearly shown to the user.

#### **Main Screen**

The main screen of the system as shown in figure 3.6, It was designed to ensure that the user could easily navigate from one section of the system to another as easily as possible. The most important activities that are conducted by the user were identified and were made accessible from the main screen. Less frequent activities are accessible from the menus. To save space on the screen, it was decided to add pictorial representations of the activity instead of text. Picture box and tool tip were used to show descriptions of the activity when the mouse hovers over the button.

A tooltip is alternatively referred to as a balloon, help balloon, or ScreenTip, a Tooltip is a tip given to an object when the mouse cursor is placed over it [7].

This was done as many classes in the system have similar operations, e.g. class Books has operation Search, Magazines, Members... etc. It also helps to increase the usability of the system.



**Figure 3.6: Main Screen of the system**

The above, figure 3.6 illustrates the design of main screen of the proposed system.

### Action Confirmation

Whenever the user clicks on a Delete or Update button, one of these confirmation boxes as shown in figure 3.7 (or any confirmation box with class specific message) pops up, asking the user to confirm this action. This is to prevent the user from accidentally editing or deleting data from the database.

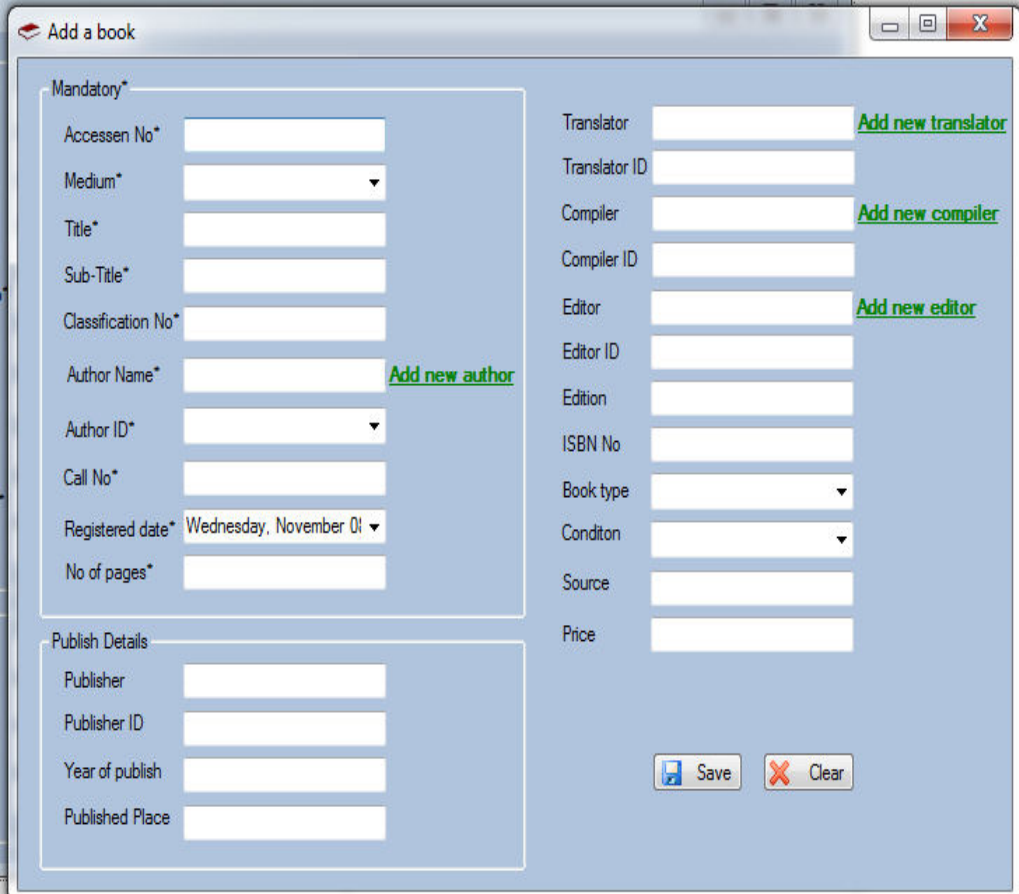


**Figure 3.7: Confirm Update, Delete Dialogs**

The above, figure 3.7 illustrates the delete and update dialog boxes of the proposed system.

## Add Books

In the form shown in figure 3.8, and many of the edit, add forms in this project, Compulsory fields are marked with the ‘\*’ mark.



The screenshot shows a web form titled "Add a book" with a window title bar. The form is organized into several sections:

- Mandatory\*** section (indicated by a red asterisk):
  - Accessen No\* (text input)
  - Medium\* (dropdown menu)
  - Title\* (text input)
  - Sub-Title\* (text input)
  - Classification No\* (text input)
  - Author Name\* (text input) with a green link "Add new author" to its right.
  - Author ID\* (dropdown menu)
  - Call No\* (text input)
  - Registered date\* (dropdown menu showing "Wednesday, November 01")
  - No of pages\* (text input)
- Publish Details** section:
  - Publisher (text input)
  - Publisher ID (text input)
  - Year of publish (text input)
  - Published Place (text input)
- Other fields** (no asterisks):
  - Translator (text input) with a green link "Add new translator" to its right.
  - Translator ID (text input)
  - Compiler (text input) with a green link "Add new compiler" to its right.
  - Compiler ID (text input)
  - Editor (text input) with a green link "Add new editor" to its right.
  - Editor ID (text input)
  - Edition (text input)
  - ISBN No (text input)
  - Book type (dropdown menu)
  - Conditon (dropdown menu)
  - Source (text input)
  - Price (text input)

At the bottom right of the form, there are two buttons: "Save" (with a floppy disk icon) and "Clear" (with a red 'X' icon).

*Figure 3.8: Add Book Form*

The above, figure 3.8 illustrates a new book registration form of the proposed system.

# CHAPTER 4 : IMPLEMENTATION

## 4.1 Introduction

Implementation means process of converting the system specification into an executable system. Design and implementation processes transform the specification to an executable program, which are, most of the time interleaved.

A well known programming language and suitable tools were selected in the process of development and coding. The codes were written and arranged according to the programming ethics which will be helped to maintain the program and further development in future.

## 4.2 Development Tools

### **Visual Studio 2008**

Taking the time into the account the time binding involved in the project, Visual Studio gave faster development with features such as IntelliSense which is an auto complete system that enables to discover and use parameters, classes...etc. quickly. Visual Studio also provides a vast range of drag and drop controls that make development simple and quick. IntelliSense is a great feature that can significantly increase one's outcome. It is made to make the development of your application much easier by automatically generate code in the Code Editor [8].

### **.NET Framework**

The .NET framework is a software development framework from Microsoft. It offers a controlled programming background where software can be developed, installed and executed on Windows-based operating systems. IT allows developers to build and deploy applications quickly. It consists of the ordinary language runtime and .NET class library.

### **Microsoft SQL Server 2008**

Microsoft SQL Server 2008 is optimized as the database server for this project. It uses Transact – SQL which is an extension to SQL. SQL Server Management Studio, which enables a set of tools for development, administration of SQL server objects, was used for development of the queries and the database as a whole.

### 4.3 Code features

The interface section of the code includes all the forms that the user interacts with the system. The inputs provided by the client are captured and sent to the base classes. To make sure the correct input was given, validation was done on the input control in the interfaces.

The bellow given code fragment shows the validation of a telephone number

```
'Telephone number validation

Private Sub txtTele4n_Leave(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTele4n.Leave
    If Not IsNumeric(txtTele4n.Text) Then
        MsgBox("Please check your Phone number!",
MsgBoxStyle.Exclamation, "LMS")
        txtTele4n.Focus()
    ElseIf txtTele4n.TextLength <> 10 Then
        MsgBox("Please check your Phone number!",
MsgBoxStyle.Exclamation, "LMS")
        txtTele4n.Focus()
    ElseIf Mid(txtTele4n.Text, 1, 1) <> "0" Then
        MsgBox("Please check your Phone number!",
MsgBoxStyle.Exclamation, "LMS")
        txtTele4n.Focus()
    End If
End Sub
```

The following code fragment shows fill the book Lending's data into the dataset

```
Private Sub Lending_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    SqlDataAdapter1.Fill(DataSet11, "Lent")
    SqlDataAdapter2.Fill(DataSet11, "Book")
    SqlDataAdapter3.Fill(DataSet11, "Member")
    SqlDataAdapter4.Fill(DataSet11, "Reservations")
    dt = DataSet11.Tables("Lent")

End Sub
```

```
'to update in Lent table
```

```
dr = dt.NewRow
dr(0) = autonno()
dr(1) = txtMemID.Text
dr(2) = txtMemName.Text
dr(3) = txtAccNo.Text
dr(4) = txtTitle.Text
dr(5) = cmbBookCondition.Text
dr(6) = dtpFrom.Value.Date
```

```
dr(7) = dtpTo.Value.Date
dt.Rows.Add(dr)
SqlDataAdapter1.Update(DataSet11, "Lent")
MsgBox("Saved")
```

The following code snippet shows to display report in Crystal Report Viewer

```
Private Sub MembersReport_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    SqlDataAdapter1.Fill(DataSet11, "Book")
    Dim cr As New CrystalReportBook
    cr.SetDataSource(DataSet11)
    CrystalReportViewer1.ReportSource = cr
End Sub
```

# CHAPTER 05: EVALUATION

## 5.1 Requirement for a good system evaluation

The system evaluation “evaluates the performance, from the view of both developers and users, of complicated systems of hardware and software. Modern computer-based information systems have become more and more complicated because of distributed computing and heterogeneous databases, and the requirement of storing large quantities of data. People are depend more and more on computer systems to support daily activities. When these systems fail, major breakdowns may occur” [9]. Therefore, in order to avoid these kinds of unnecessary failures, a good system evaluation and testing has to be introduced.

## 5.2 Testing

System Testing is the testing of a entire and fully integrated software product. Generally software is only one element of a larger computer based system. In the end, software is interfaced with other software/hardware systems. System testing is actually a sequence of different tests whose sole purpose is to exercise the full computer based system. The test outcome help a developer to spot and reduce the errors (software bugs) which may arise from the actual working environment of the system.

## 5.3 Test plan

The test Plan of the Library Management System defines the testing strategies and the approaches to testing. Quality assurance will be used to certify the quality of this system prior to its using in the actual environment. It also contains various resources essential for the successful completion of this project. When coding the Library Management System, a unit testing was carried out for the functions and procedures to ensure that they perform as anticipated. This is a code level testing which tests the units of codes [10]. An integration testing has been done in order to test the modules (integration of units of code) of the system. This is important when using different kinds of implementing technologies simultaneously. It ensures that all the integrated components are running cooperatively for the intended result.

After the completion of unit and integration testing the whole system has been tested correctly. The system testing guarantees the standard of the system. A system testing was conducted using a similar environment where the system will be really used [11]. After completing the system a confirmation has been obtained from the clients through a user acceptance testing whether their requirements are fulfilled by the developed system.

Further the test plan includes two levels for implementing the “Black box” and “White box” testing methods. The Black box testing focuses on the functions and behaviors of the system. This testing does not need any knowledge on the internal structure of the system. The White box testing focuses the internal structure and logics of the system. This is a code level testing method and any malfunctioning units can be easily detected [12].

## 5.4 Test Case Generation

The most essential things of any test plan are the test cases. Generally test cases include the test case title, inputs, expected results and the priority. In order to diminish the complication, the whole system has been divided in to several modules, and each module was tested separately. The tables 5.1 to 5.5 show a few of the test cases used to conduct testing. The rest of the test cases can be found in Appendix E– Test Results.

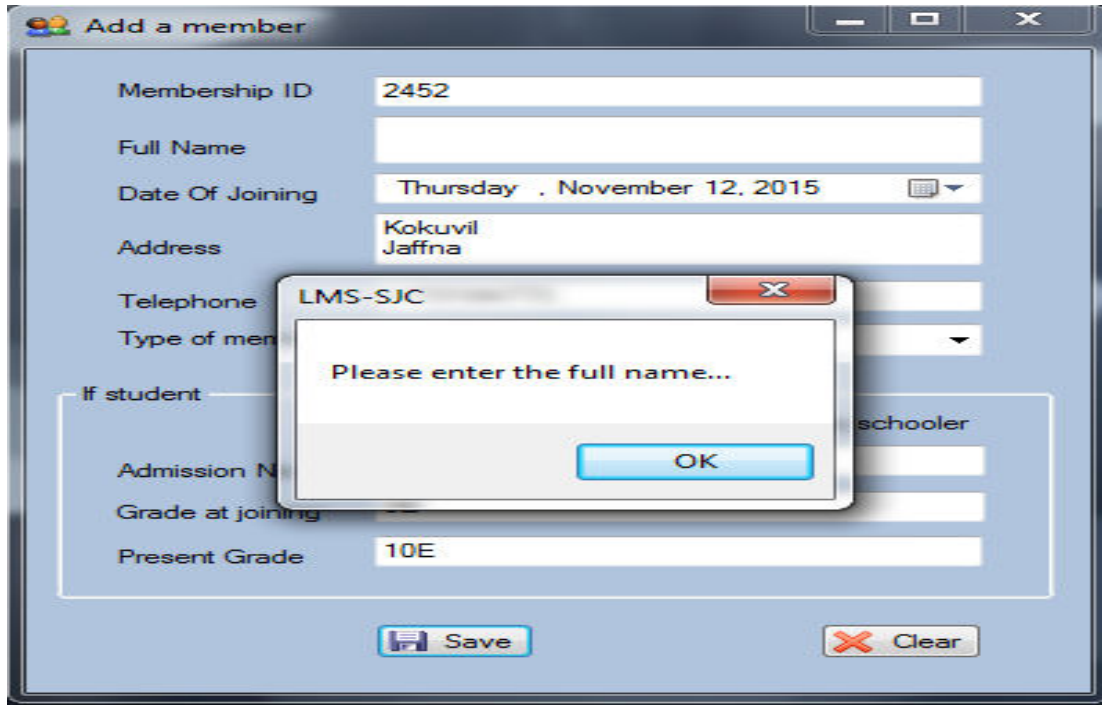
### Register Members

No	Test Case	Expected Output	Actual Output	Status
01	Not filling all required fields.	Registration is not allowed	Registration is not allowed	Pass
02	All required fields are filled.	Registration is made and Membership Number is displayed.	Registration is made and Membership Number is displayed.	Pass
03	When member already exists	Shows error message	Shows error message	Pass
04	Non numeric values are entered into the Telephone and Receipt fields	Show error message and clear field	Show error message and clear field	Pass
05	Registering staff members.	Parent, Receipt No, Class and Grade fields are disabled.	Parent, Receipt No, Class and Grade fields are disabled.	Pass

**Table 5.1: Register Members Test Results**

The above, table 5.1 describes member registration test case results of the proposed system.





**Figure 5.1: Member registration Test Result**

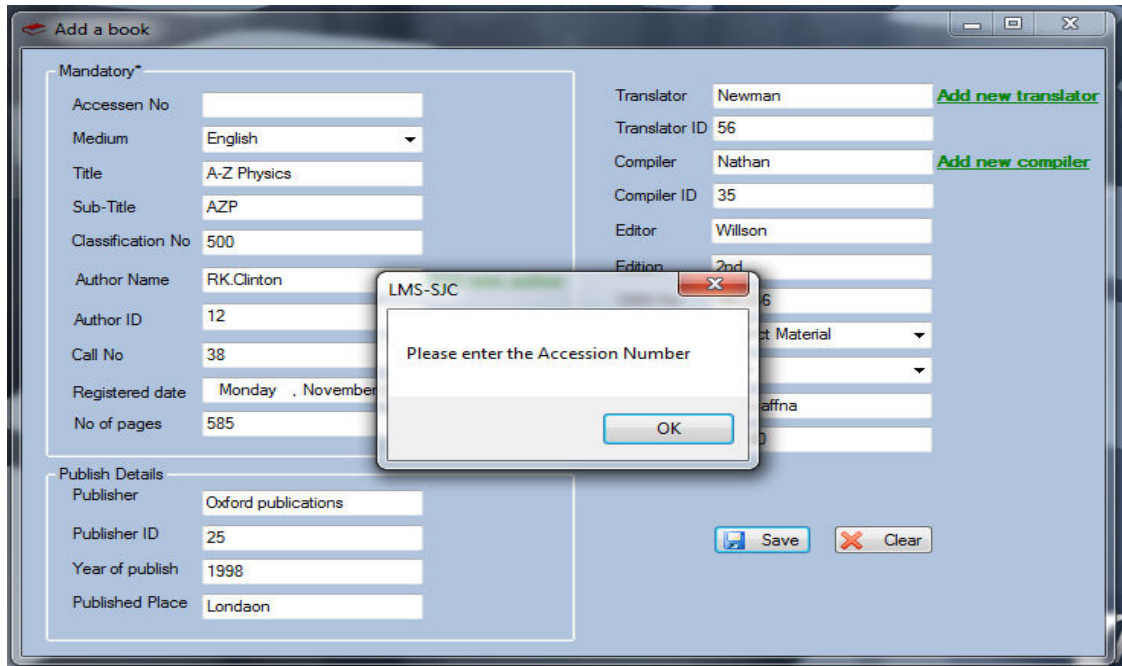
The above, figure 5.1 illustrates member registration test case results of the proposed system.

### Add books

No	Test Case	Expected Output	Actual Output	Status
01	When required fields are not entered	Prompts user to enter required fields	Prompts user to enter required fields	Pass
02	Acquisition Date is in the future	Prompts the user to select the current or past date. Sets fields to current date	Prompts the user to select the current or past date. Sets fields to current date	Pass
03	All required fields are entered	Save book	Save book	Pass
04	Pages and Price are not numeric	Prompt user to enter numeric values.	Prompt user to enter numeric values.	Pass

**Table 5.2: Add Books Test Results**

The above, table 5.2 describes add books test results of the proposed system.



**Figure 5.2: Book Details Test Result**

The above, figure 5.2 illustrates add books test results of the proposed system.

### Delete books

No	Test Case	Expected Output	Actual Output	Status
01	Delete Books	Prompt user for Confirmation	Prompt user for Confirmation	Pass
		Delete book from catalogue, as well as author publisher where they have 0 books	Delete book from catalogue, as well as author publisher where they have 0 books	Pass
02	Delete Author	Relevant, Author, books and publishers should be deleted	Relevant, Author, books and publishers should be deleted	Pass
03	Delete Book Publisher	Publisher and relevant book and Author should be deleted	Publisher and relevant book and Author should be deleted	Pass

**Table 5.3: Delete Books Test Results**

The above, table 5.3 describes delete books test case results of the proposed system.

## Magazines

No	Test Case	Expected Output	Actual Output	Status
01	Delete Magazine Issue	Deletes magazine issue, and magazine, magazine publisher where only one magazine is available for each	Deletes magazine issue, and magazine, magazine publisher where only one magazine is available for each	Pass
02	Delete Magazine	Delete Magazine, magazine issues and magazine publisher	Delete Magazine, magazine issues and magazine publisher	Pass
03	Delete publisher	Should delete publisher, magazine and any magazine issues	Should delete publisher, magazine and any magazine issues	Pass

**Table 5.4: Delete Magazines Test Results**

The above, table 5.4 describes delete magazines test case results of the proposed system.

## Lending Books

No	Test Case	Expected Output	Actual Output	Status
01	Searched Book found	Book information shown in form	Book information shown in form	Pass
02	Searched Member found	Member information is shown in the form	Member information is shown in the form	Pass
03	Searched Book is not found	Error is shown in the form, Lend button is disabled	Error is shown in the form, Lend button is disabled	Pass
04	Searched Member is not found	Error is shown in the form, Lend button is disabled	Error is shown in the form, Lend button is disabled	Pass
05	Member and Book is found, but Book is reserved	Show failure of lend process	Show failure of lend process	Pass
06	Member and book is found and the book is not reserved	Show success of lend process	Show success of lend process	Pass

**Table 5.5: Lending Books Test Results**

The above, table 5.5 describes Lending books test case results of the proposed system.

## User Acceptance Testing.

The system was tested by the Librarian to see if all the functional and non functional requirements of the system were met. The librarians response showed that the all the anticipated system modules were implemented and that she was happy with the system that was built.

# CHAPTER 06: CONCLUSION

## 6.1 OVERVIEW

Along the various stages of this project, whatever work done, was checked along with the client requirements to make sure that those requirements have been addressed during those phases. This steady checking with the requirements made sure that the developed system met the goals and objectives that were planned at the beginning of the project.

By reviewing the functional and non functional requirements that were discovered during the analysis phase and checking back with the functionalities enforced in the developed system, it can be said that all the requirements of the user have been satisfied.

The simple and intuitive user interface that was designed and developed was easy to learn and use proved to be satisfactory for the user.

The reservations module and fine module are new facilities which the existing system does not have. This was because there was no quick way to find out if a book was borrowed or not, without checking all the membership cards and referring the books register. The built system allows the Librarian to search a book quickly and then check its availability.

The ability to check overdue books quickly allowed the Librarian to make sure that the members have returned all borrowed books before applying for new membership cards. The system alleviates the risk of losing books this way.

## 6.2 Lessons Learnt

This developed system does not merely fulfill the requirement of the final year of the Degree program; however it assists me to practically apply the knowledge learnt throughout the previous years.

When handing over the project proposal, I did not have much of an idea on how to do the project. When progressing through step by step according to the guideline provided by the university, I gained a precious knowledge on how to do a successful professional system development project. By doing the development process according to a schedule, I learnt how to do my day-to-day activities by managing time efficiently.

The implementation phase was the difficult and most interesting phase of the project, as it let me to try out practically the academic knowledge that I have gained on programming languages such as SQL and the tools available in the Visual Studio IDE and the SQL Server Management Studio.

Writing the dissertation was another interesting task of the project. It provided me with lessons on how to write a report in a professional manner. It helped me to develop my skills on writing and designing technical reports.

### 6.3 Future Work

While the current system allows addition of new data into the system, it does not grant an easy method to add historic data into the database. Adding books manually, using the provided insertion point is time consuming. Since an excel file containing the information of the book already exists it can perhaps be used to automate the addition of books into the database.

The system could in the future be connected to an employee management system and student management system. This could make member registration faster and more efficient as there would be a central database where all the member information is stored and accessed using the Library Management System. This will result in the Librarian only having to enter the admission or staff number to register a member.

# REFERENCES

- [1]<http://www.koha.org/about> [ Accessed: 8 February,2017]
- [2]<https://socialhistory.org/en/projects/evergreen-integrated-library-system> [ Accessed: 8 February,2017]
- [3] <https://www.techrepublic.com/blog/10-things/10-techniques-for-gathering-requirements/> [ Accessed: 8 February,2017]
- [4] <https://www.simplilearn.com/feasibility-study-article> [ Accessed: 13 March,2017]
- [5] <https://www.slideshare.net/jagaarj/database-design-normalization> [ Accessed: 21 April,2017]
- [6] <https://app.creately.com/diagram/vector/hwj2pb8q1> [ Accessed: 05 May,2017]
- [7]<https://www.computerhope.com/jargon/t/tooltip.htm> [ Accessed: 17 June,2017]
- [8] <http://www.developerfusion.com/article/7633/introducing-visual-studio-net-2008-top-10-features/> [ Accessed: 23 July,2017]
- [9] <http://ctb.ku.edu/en/table-of-contents/evaluate/evaluation/evaluation-plan/main> [ Accessed: 6 August,2017]
- [10] <http://istqbexamcertification.com/what-is-unit-testing/> [ Accessed: 14 August,2017]
- [11][https://www.tutorialspoint.com/software\\_testing\\_dictionary/integration\\_testing.htm](https://www.tutorialspoint.com/software_testing_dictionary/integration_testing.htm) [ Accessed: 26 September,2017]
- [12]<https://www.testing-whiz.com/blog/understanding-white-box-testing-and> [ Accessed: 08 October,2017]

# Appendix A – System documentation

## System Manual

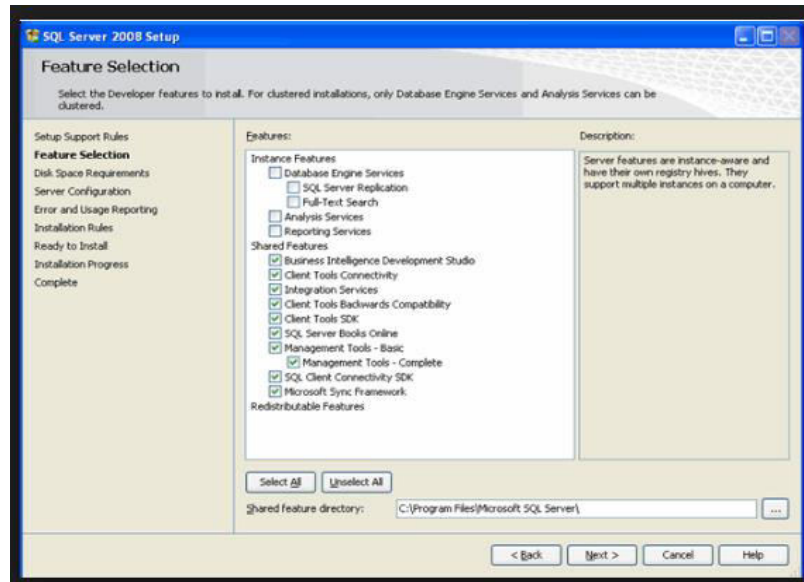
### Step 1: Install .NET Framework 3.5



**Figure A.1: Install .NET Framework**

The above, figure A.1 illustrates installation step of .NET Framework

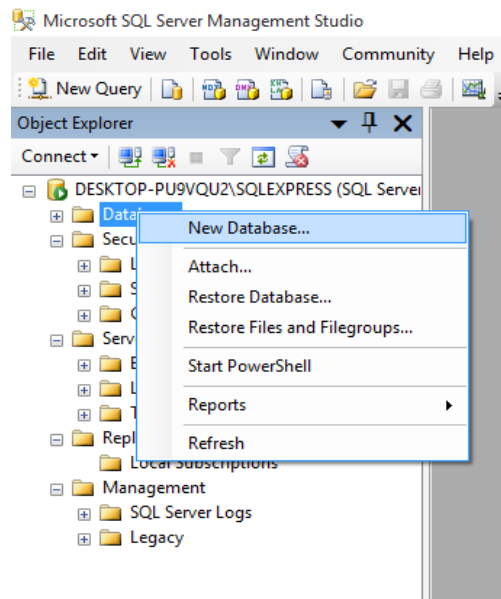
## Step 2: Install SQL Server 2008



*Figure A.2: Install SQL Server 2008*

The above, figure A.2 illustrates installation step of SQL Server 2008

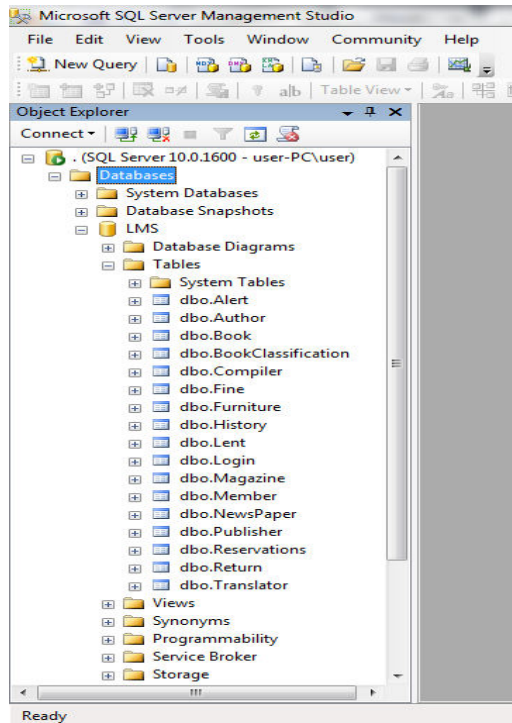
## Step 3: Create Database 'LMS' in the server.



*Figure A.3: Create New Database*

The above, figure A.3 illustrates creating a new data base in SQL Server 2008

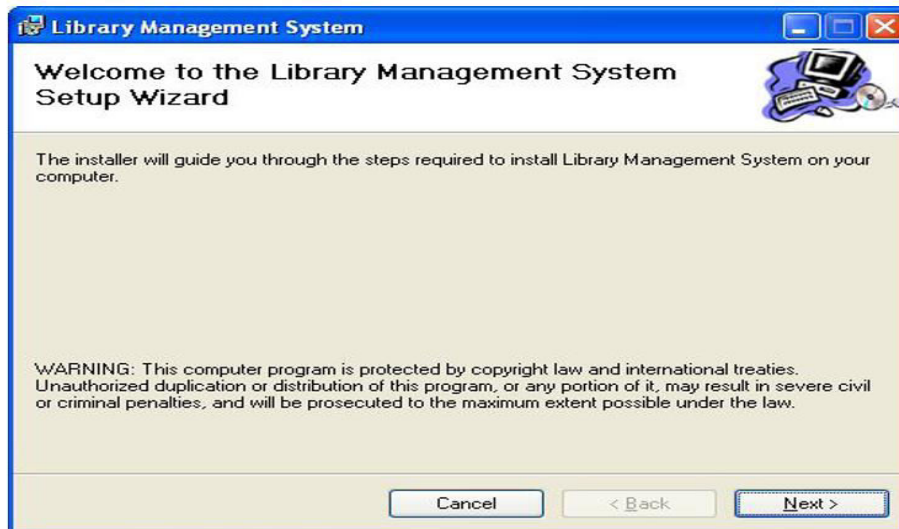




**Figure A.4: Database Created**

The above, figure A.4 illustrates created database(LMS) in SQL Server 2008

**Step 4:** Double click the Library Management System setup file from the setup media. The installer will start and Library Management System will install on your computer.



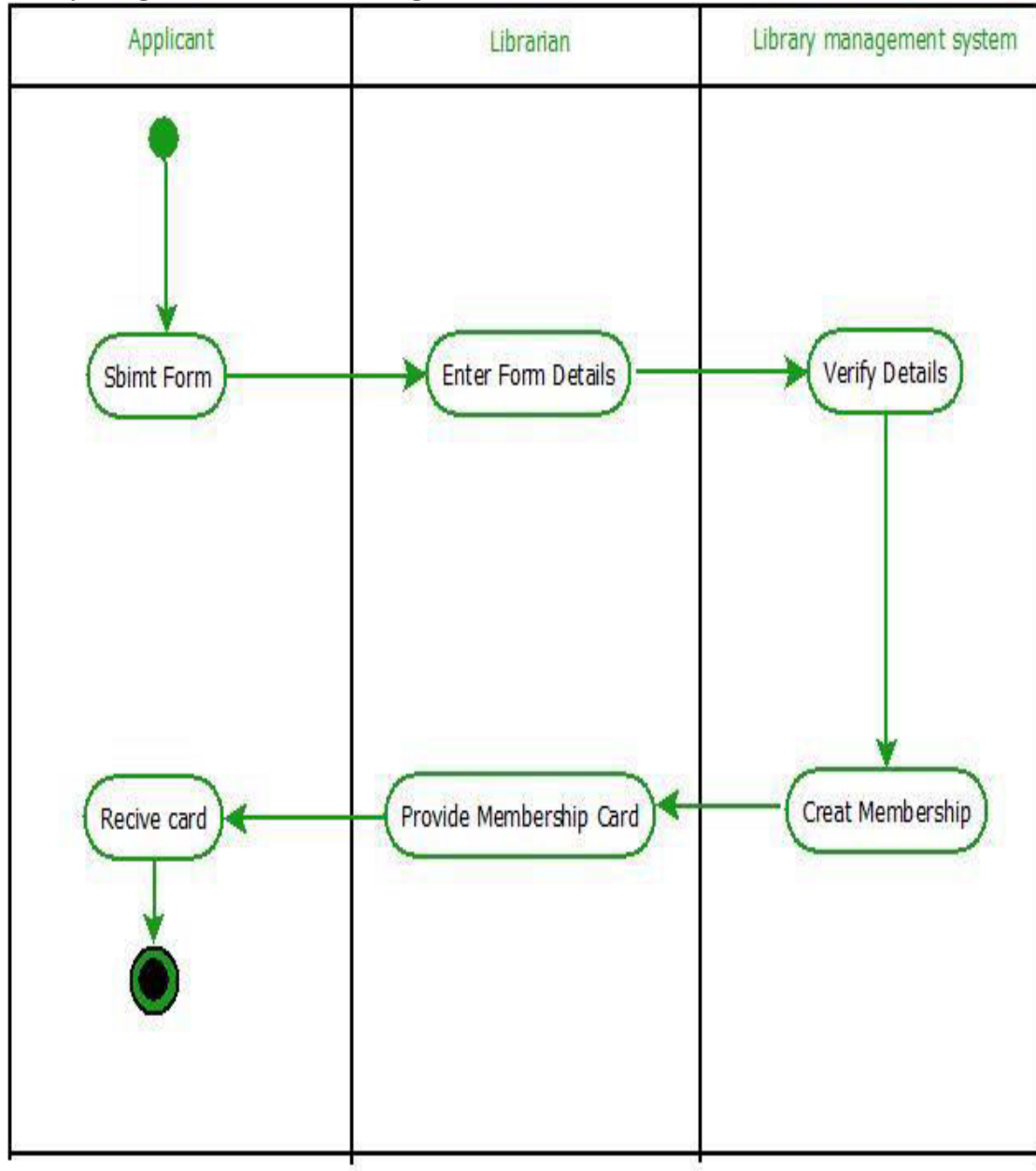
**Figure A.5: Install Application**

The above, figure A.5 illustrates installation of developed software.

Follow the instructions and the program will install.

# Appendix B – Design Documentation

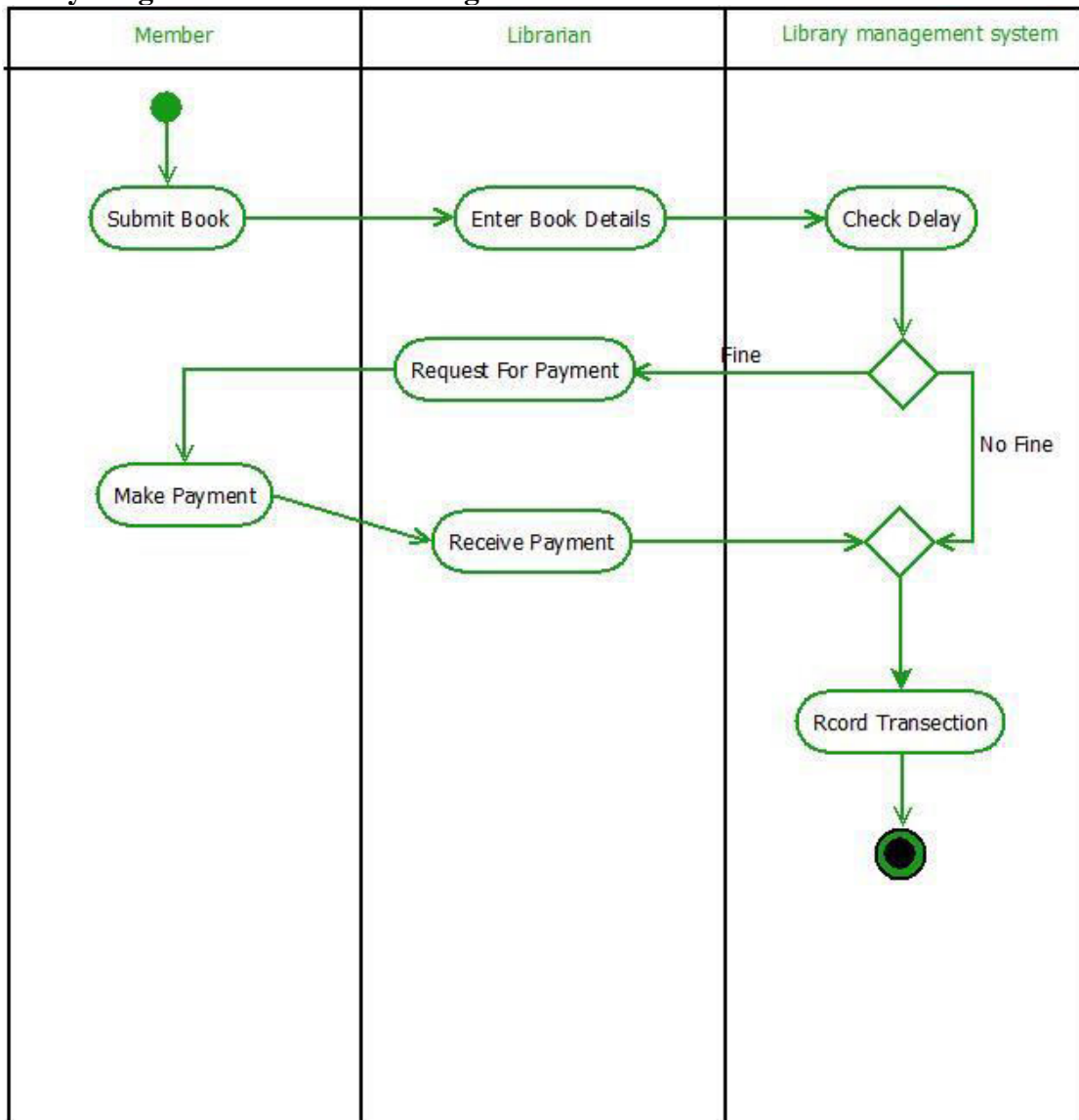
## Activity Diagram for Member Registration



**Figure B.1: Activity Diagram for Member Registration**

The above, figure B.1 illustrates the activity diagram of member registration.

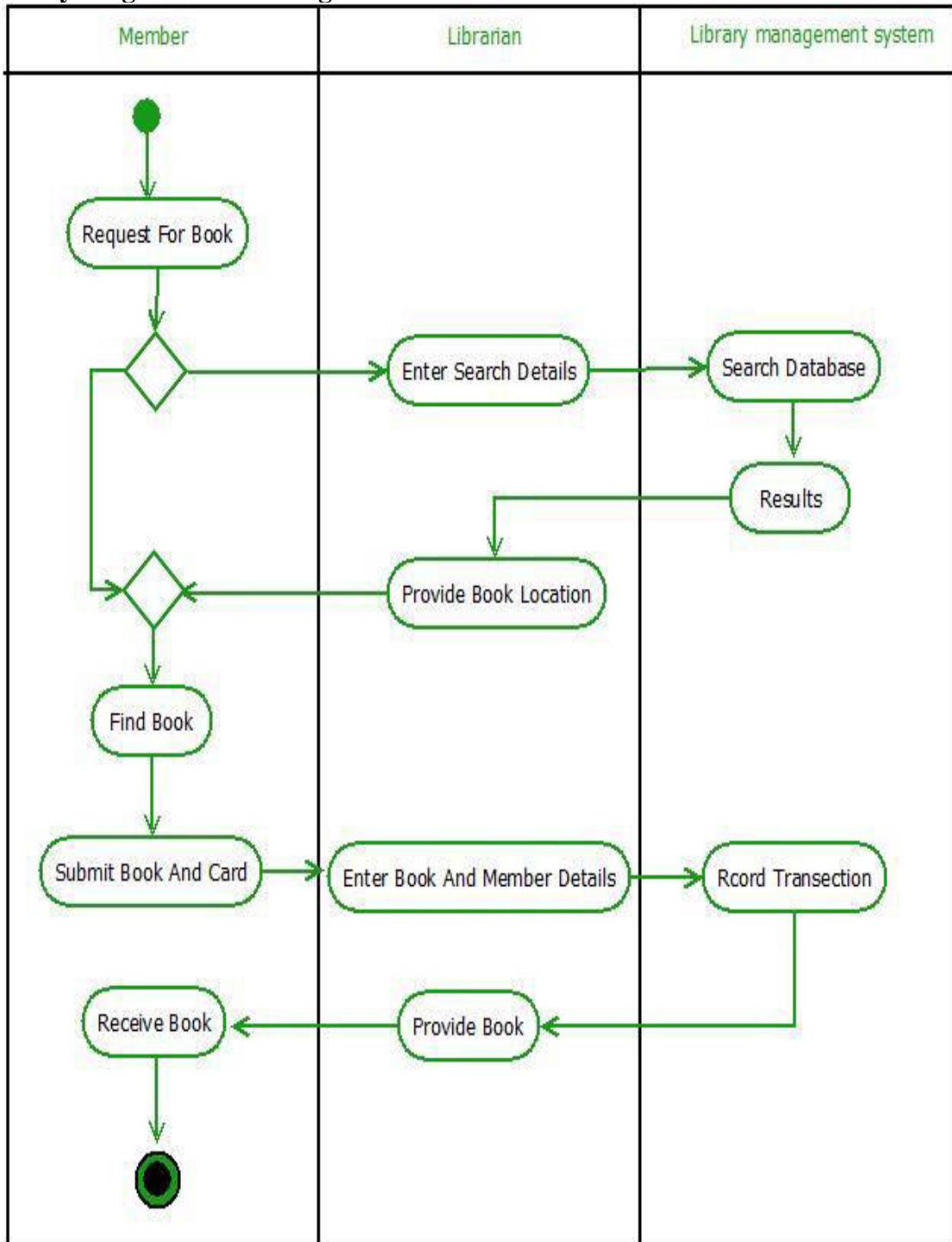
### Activity Diagram for Book Returning



**Figure B.2: Activity Diagram for Book Returning**

The above, figure B.2 illustrates the activity diagram of book return.

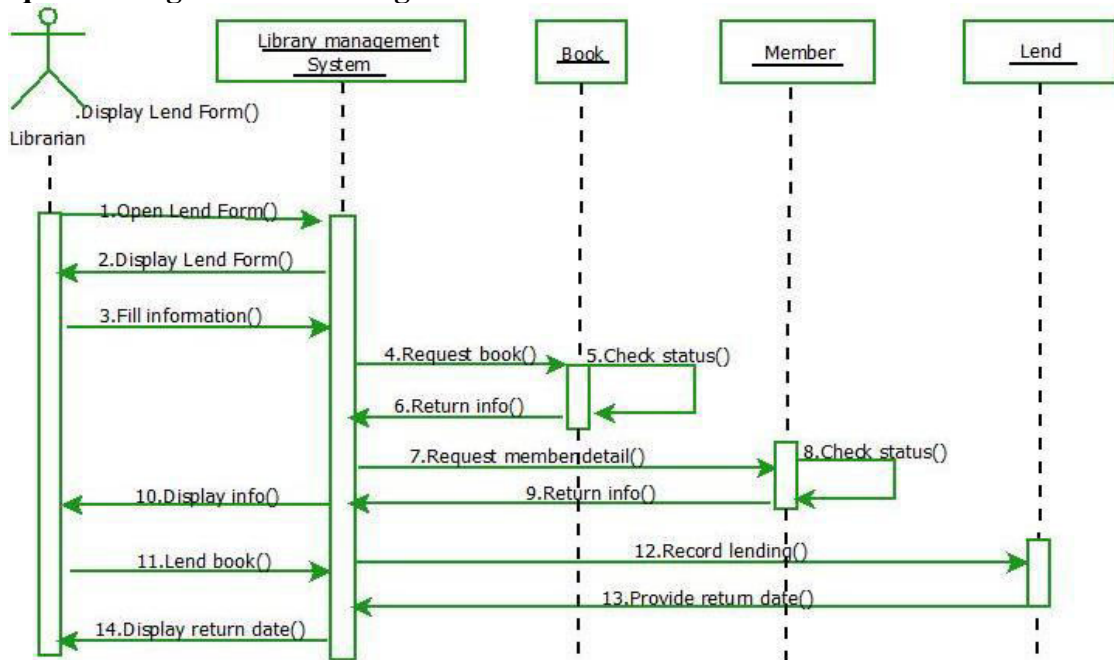
### Activity Diagram for Lending Book



**Figure B.3: Activity Diagram for Lending Books**

The above, figure B.3 illustrates the activity diagram of lending.

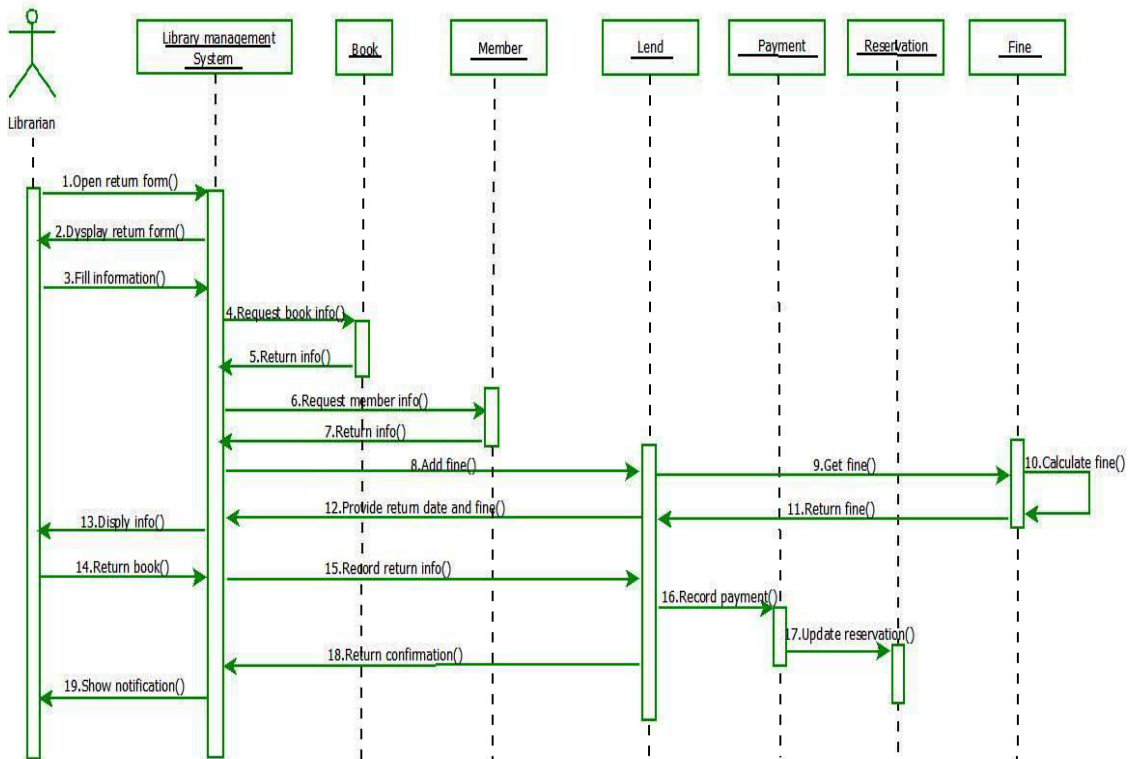
## Sequence Diagram for Lending a Book



**Figure B.4: Sequence Diagram for Lending a Book**

The above, figure B.4 illustrates the sequence diagram of lending.

## Sequence Diagram for Returning a Book

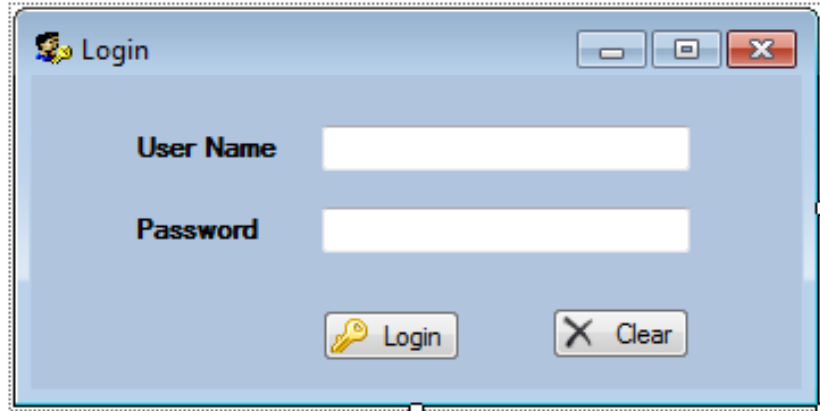


**Figure B.5: Sequence Diagram for returning a book**

The above, figure B.5 illustrates the sequence diagram of book return.

# Appendix C – User Documentation

## Login Screen



*Figure C.1: Login Screen*

The above, figure C.1 illustrates the design of login screen of proposed system.

This is the Login screen, which appears at startup. The user is required to enter his/her login information. If you have entered the wrong username or password, you will not be allowed to access the system.

## Main Screen



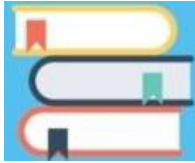
*Figure C.2: Main Screen*

The above, figure C.2 illustrates the design of login screen of proposed system.

This is the main screen of the system. Here you will find the main activities of that are carried out in the library.

## Activities

The following are the activities you can navigate to from the main screen.



Add a new book



Add a new member



Add a new furniture



Lend a book



Return a book



Reserve a book



Do fining

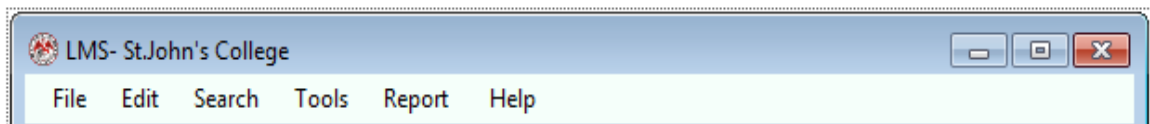


**Figure C.3: Main Tasks**

The above, figure C.3 illustrates the icons of the main tasks of proposed system.

Apart from these a few secondary activities can be navigated to from the menus.

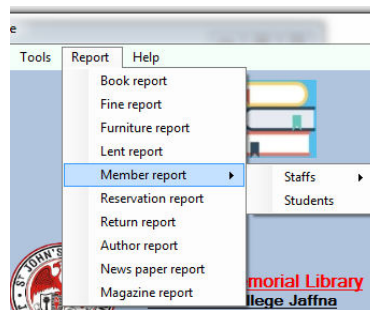
### Menu Bar



**Figure C.4: Menu Items**

The above, figure C.4 illustrates the menu bar of proposed system.

### Report Submenu Items

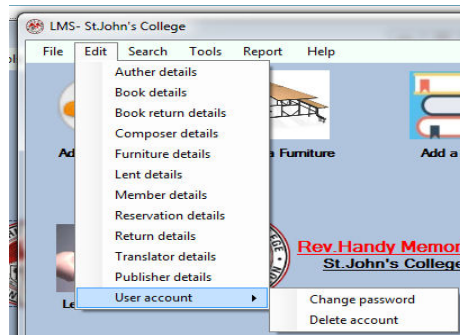


**Figure C.5: Report Submenu Items**

The above, figure C.5 illustrates the sub-menu items of report.



## Edit Submenu Item



**Figure C.6: Edit Submenu Items**

The above, figure C.6 illustrates the sub-menu items of edit.

## Member Registration

To register members click the member button on the main toolbar. This will bring up the members dialog box.

A screenshot of a web application window titled "Add a member". The form contains several input fields: "Membership ID", "Full Name", "Date Of Joining" (with a calendar icon and the value "Tuesday, November 07, 2017"), "Address", "Telephone", and "Type of member" (a dropdown menu). Below these is a section for "If student" with two radio buttons: "Boader" and "Day schooler". Further down are three more input fields: "Admission No", "Grade at joining", and "Present Grade". At the bottom of the form are two buttons: "Save" and "Clear".

**Figure C.7: Register Members Form**

The above, figure C.7 illustrates member registration form.

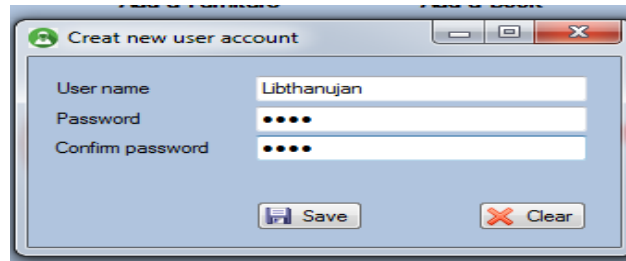
You can select the type of user to register. The success and the failure of the action will be shown in the message box.

## User Accounts

There are two types of user accounts. One is the Librarian (Admin) and the other is the User. In terms of the Library Management System, the only difference between these two accounts is that the librarian is allowed to alter any accounts and do all task (add books, member, magazine,...Etc). The user can only view the catalogue and search information.

### Create New User

Click on the Create new user account at File menu item.

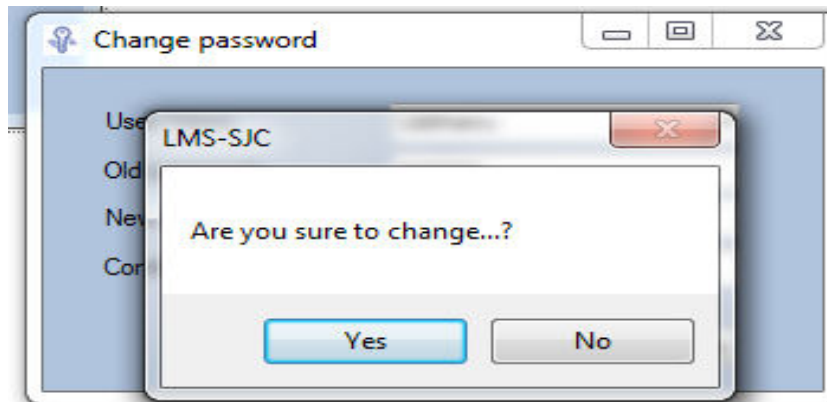


*Figure C.8: Create new user account form*

The above, figure C.8 illustrates the new user account creation form.

### Change Password for User Account

To change password, want to select user name and enter that password, if it is right then can change new password to that user

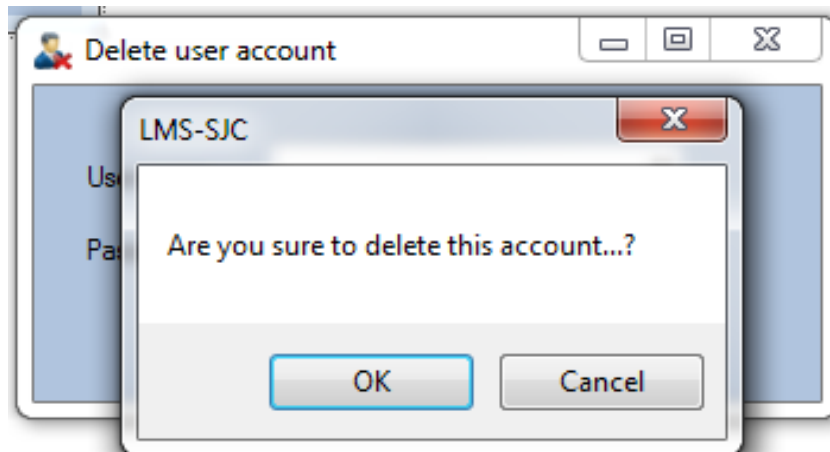


*Figure C.9: Change Password Form*

The above, figure C.9 illustrates the password changing form.

## Delete User

Here, can delete user role only. Can't delete admin role.

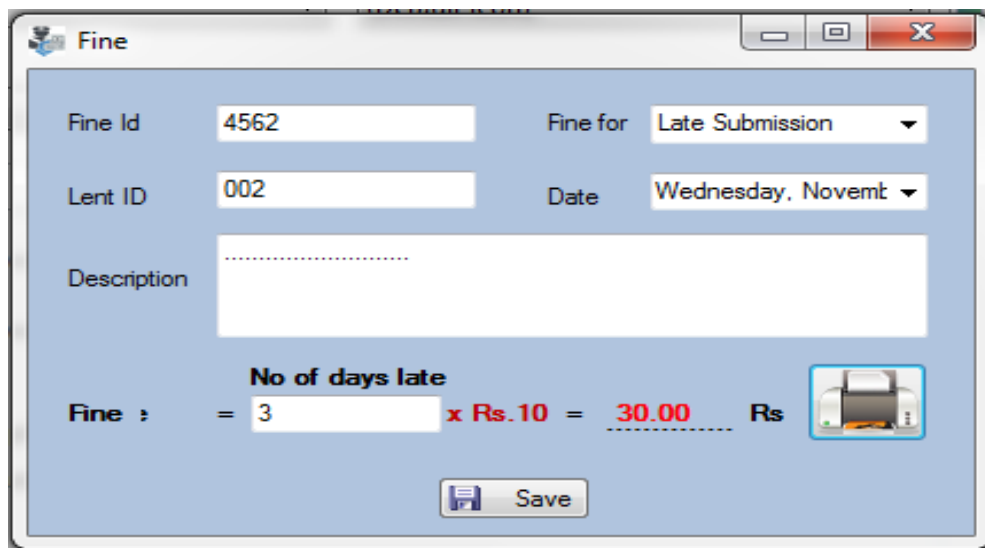


*Figure C.10: Delete User Account Form*

The above, figure C.10 illustrates delete user account form.

## Fine calculation

Fine calculation is done by obtaining the return date of the book and current date. The dates between these two dates are checked and fine will be calculated.



*Figure C.11: Fine Form*

The above, figure C.11 illustrates the design of fine calculation form.

When the return button is pressed, the payment screen will show up with the payment already added to the database.

## Return book

The image shows a software window titled "Return" with a green arrow icon on the left and standard window controls (minimize, maximize, close) on the right. The window contains a form with the following fields and controls:


- Lend ID:** A dropdown menu.
- Accession No:** A text input field.
- Lent date:** A date picker showing "Wednesday, November 01, 2017".
- Return Date:** A date picker showing "Wednesday, November 08, 2017".
- Book Condition:** A dropdown menu.
- No of days late:** A text input field.
- GO TO FINE:** A red text link.
- Save:** A button with a floppy disk icon.
- Clear:** A button with a red 'X' icon.

*Figure C.12: Return Form*

The above, figure C.12 illustrates design of book return form.


# Appendix D – Management Reports

## Books Details Report

		<b>Rev.C.C.Handy Memorial Library</b>		
<b>Books Details Report</b>				
11/7/2017				
<u>AccessionNo</u>	<u>Title</u>	<u>Medium</u>	<u>Price</u>	<u>Source</u>
00001	Holly Bible	English	2,000.00	Youth for christ Jaffna
00002	The Merchant Of Veni	English	1,500.00	OBA1996
00003	Advanced Level Phys	English	3,850.80	School
00006	Aathisudi	Tamil	850.00	OBA 1985
00007	Ramayanam	Tamil	850.00	USAID
00875	Tourism & Hospitality	English	2,852.00	School
01205	Kalaikalanjiyam	Tamil	3,000.00	Mr.Jude (London)
84510	Kathal Malai	Tamil	250.00	School
<b>Total Books</b>				<b>8</b>


*Figure D.1: New Books Acquisition Report*

## Member Details Report

		<b>Rev.C.C.Handy Memorial Library</b>					
<b>Member Details Report</b>							
11/7/2017							
<u>MembershipId</u>	<u>Fname</u>	<u>Jointdate</u>	<u>CurrentGrade</u>	<u>Address</u>	<u>ContactNo</u>	<u>MemCategor</u>	<u>StudentType</u>
0001	Daniel Nitharshan	10/6/2017		Nallur Jaffna	0772055224	Accadameic	
0002	Christian Angelove	8/17/2008	12	Chundikuli Ja	0755184773	Student	Day schooler
0003	Raja Niroshan	2/6/2014	11	Racka Road J	0213214151	Student	Boarder
0004	Tharmakulasingam Tharmer	1/1/2017	14	Thirunellveli J	0779785796	Student	Day schooler
0005	Ajanthini Gnanenthiran	4/14/2000		Ariyalai Jaffna	0774935479	Accademic st	
0006	Sirippu Thanusiya	4/11/2015		Ariyalai	0774846421	Non Acadami	
<b>Grand Total:</b>							<b>6</b>


*Figure D.2:Members Report*

## Furniture Details Report

 <b>Rev.C.C.Handy Memorial Library</b> <b>Furniture Details Report</b>						
11/7/2017						
<u>FurnitureId</u>	<u>Furniture</u>	<u>Furnituretype</u>	<u>Brand</u>	<u>Condition</u>	<u>IssuedDat</u>	<u>Source</u>
0005	Rack	Rack(Wooden)	Composed	New	2/14/2008	OBA1999
0006	Computer	Laptop	HP	Used	2/8/2016	School
0018	Wall Clock	Wall Clock(Metal)	Taitan	New	10/3/2000	Mr.G.Wilson 1970 batch
98797	Almirah	Armchair(Wooden)	Piyesta	New	11/3/2017	OBA 2005
CH0001	Chair	Armlesschair(Wooden	Damro	New	10/23/2011	MrA..Jason 1995 batch
<b>Total Furnitures</b>						<b>5</b>


*Figure D.3: Furniture Details Report Details*

## Author Report

 <b>Rev.C.C.Handy Memorial Library</b> <b>Book-Author Details Report</b>	
11/7/2017	
<u>Author Id</u>	<u>Author Name</u>
1	William Shakespeare
2	Gideons
3	K.Daniel
4	Kalki
5	Kannathasan
6	Subramaniyaparathiyar
7	Nelkon and Parker
8	Wren and Marteen
9	Ernest Hemingway
10	Auvayaar
11	Joseph
<b>Grand Total:</b>	<b>11</b>


*Figure D.4: Author Report*

## Magazine Report

 <b>Rev.C.C.Handy Memorial Library</b> <b>Magazine Details Report</b>		
11/7/2017		
<u>MagazineId</u>	<u>Name</u>	<u>Medium</u>
1	Times	English
2	Kalaikesari	Tamil
3	Explor Sri Lanka	English
4	The Sri Lankan Scientist	English
5	Kungumam	Tamil
6	Ananda Vikadan	Tamil
7	Puthiya Thalaimurai	Tamil
<b>Total Magazine</b>		<b>7</b>

*Figure D.5: Magazine Report*

## Lending Report

 <b>Rev.C.C.Handy Memorial Library</b> <b>Lending Details Report</b>							
11/7/2017							
<u>LentId</u>	<u>MemberID</u>	<u>AccessionNo</u>	<u>Title</u>	<u>From</u>	<u>To</u>	<u>Name</u>	<u>BookCondition</u>
001	0003	00006	Aathisudi	11/1/2017	11/1/2017	Ajai Kumar	Good
002	0001	00001	Holly Bible	11/1/2017	11/17/2017	Raj Suthan	Damaged
8787	0001	00002	The Merchant Of Venice	11/3/2017	11/10/2017	Raj Suthan	Old
<b>Total Lendings</b>							<b>3</b>

*Figure D.6: Lending Report*

# Appendix E - Test Results

## Add Magazines

No	Test Case	Expected Output	Actual Output	Status
01	Not filling all required fields.	Prompts user to enter required fields	Prompts user to enter required fields	Pass
02	Acquisition Date, Issue Date is in the future	Prompts the user to select the current or past date. Sets fields to current date	Prompts the user to select the current or past date. Sets fields to current date	Pass
03	All required fields are entered	Save Magazine	Save Magazine	Pass

**Table E.1: Add Magazines Test Results**

The above, table E.1 describes add magazines test results of the proposed system.

## Search Books

No	Test Case	Expected Output	Actual Output	Status
01	Search Title All – distinct values	Shows all results in grid with copies	Shows all results in grid with copies	Pass
02	Search Title using field	Shows search result filtered by the fields	Shows search result filtered by the fields	Pass
03	All required fields are entered	Save Magazine	Save Magazine	Pass
04	Search Author All	Search Author All	Search Author All	Pass
05	Search books through Author	Shows all books by that author	Shows all books by that author	Pass
06	Search Publishers All	All Shows all book Publishers	All Shows all book Publishers	Pass
07	Search Publishers using fields	Shows all book publishers filtered by the fields	Shows all book publishers filtered by the fields	Pass
08	Search books by publishers	Shows all books by the publisher	Shows all books by the publisher	Pass

**Table E.2: Search Books Test Results**

The above, table E.2 describes book search test results of the proposed system.



### Magazines

No	Test Case	Expected Output	Actual Output	Status
01	Search Magazine - All	Shows all magazines	Shows all magazines	Pass
02	Search Magazines using fields	using fields Shows magazines filtered by fields	using fields Shows magazines filtered by fields	Pass
03	Search Magazine Publishers	Shows all magazine Publishers	Shows all magazine Publishers	Pass
04	Search Magazine through magazine publishers	Shows all magazine selected by publishers	Shows all magazine selected by publishers	Pass

**Table E.3: Search Magazines Test Results**

The above, table E.3 describes magazine search test results of the proposed system.

### Search Members

No	Test Case	Expected Output	Actual Output	Status
01	Search all members	Display all members	Display all members	Pass
02	Search members using fields	shows members filtered by fields	shows members filtered by fields	Pass
03	irrelevant search fields are disabled when searching staff members	Fields class, grade are disabled	Fields class, grade are disabled	Pass

**Table E.4: Search Members Test Results**

The above, table E.4 describes search members test results of the proposed system.

### Edit Member

No	Test Case	Expected Output	Actual Output	Status
01	When editing staff members	fields grade, class should be disabled	fields grade, class should be disabled	Pass
02	editing members	members edited	members edited	Pass

**Table E.5: Edit Members Test Results**

The above, table E.5 describes edit members test results of the proposed system.

### Returning Books

No	Test Case	Expected Output	Actual Output	Status
01	Searched Book is not Lend.	Shows error in form	Shows error in form	Pass
02	Return Date has not passed	Fine amount is 0	Fine amount is 0	Pass
03	Return Date has passed	Expected fine amount shown	Expected fine amount shown	Pass
04	Click return button when fine in 0	Show message that books has been returned	Show message that books has been returned	Pass
05	Click return button when fine is greater than 0 Open	Open payment form with payment already added	Open payment form with payment already added	Pass

**Table E.6: Return Books Test Results**

The above, table E.7 describes return of books test results of the proposed system.

### Lending

No	Test Case	Expected Output	Actual Output	Status
01	Searched Items does not exist	Show message in form, disable lend button	Show message in form, disable lend button	Pass
02	Searched Items does exist but is borrowed or reserved	Shows message in form, disable lend button	Shows message in form, disable lend button	Pass
03	Lend button is clicked	Shows success or failure of the operation in the form	Shows success or failure of the operation in the form	Pass

**Table E.7: Lending Test Results**

The above, table E.8 describes book lending test results of the proposed system.

## User Acceptance Testing result

### User Evaluation Form for Library Management System

School Name : ST. JOHN'S COLLEGE.....  
Tester Name : MRS. THULASI RAGULARAJH.  
Tester Role : LIBRARIAN.....

No	Module	Satisfied: Yes/No
01	Member registration module	yes
02	Member details update module	yes
03	Book registration module	yes
04	Book details update module	yes
05	Furniture registration module	yes
06	Furniture details update module	yes
05	Search module	yes
06	Lend module	yes
07	Return module	yes
08	Reservation module	yes
09	Fine module	yes
10	Report module	yes

R.Thulsi  
Signature

29-09-2017  
Date

# Appendix F – Code Listing

## Code for login

```
Public Class Login
    Private Sub Login1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SqlDataAdapter1.Fill(DataSet11, "Login")
    End Sub
    Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLogin.Click
        Dim i As Integer
        For i = 0 To DataSet11.Tables("Login").Rows.Count - 1
            If txtUserName.Text =
DataSet11.Tables("Login").Rows(i).Item(0) And txtPassword.Text =
DataSet11.Tables("Login").Rows(i).Item(1) Then
                MsgBox("Welcome you as " &
DataSet11.Tables("Login").Rows(i).Item(2))
                Me.Hide()
                Main.Show()
                Exit For
            End If
        Next
        If i = DataSet11.Tables("Login").Rows.Count Then
            MsgBox("User Name or Password error!")
        End If
    End Sub
    Private Sub Clear()
        txtUserName.Text = ""
        txtPassword.Text = ""
    End Sub
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnCancel.Click
        Call Clear()
    End Sub
    Private Sub Label14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        Newuseraccount.Show()
        txtUserName.Focus()
    End Sub
End Class
```

## Coding for add books

```
Public Class Bookre
    Dim dt As DataTable
    Dim dr As DataRow
    Private Sub Bookre_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'adepter fill
        SqlDataAdapter1.Fill(DataSet11, "Book")
        SqlDataAdapter2.Fill(DataSet11, "Author")
        dt = DataSet11.Tables("Book")
        cmbAuthorId.Text = ""
    End Sub
    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
        If txtAccNo.Text = "" Then
            MsgBox("Please enter the Accession Number")
            txtAccNo.Focus()
        ElseIf txtClassfiNo.Text = "" Then
            MsgBox("Please the classification no...", , "LMS")
            txtClassfiNo.Focus()
        ElseIf txtTitle.Text = "" Then
            MsgBox("Please enter the title...",
MsgBoxStyle.Exclamation, "LMS")
            txtTitle.Focus()
        ElseIf txtAuthorName.Text = "" Then
            MsgBox("Please select the author name...", , "LMS")
            txtAuthorName.Focus()
        Exit Sub
    End If
    If MsgBox("Are you sure to Save this record...",
MsgBoxStyle.YesNo, "Save") = MsgBoxResult.Yes Then
        dr = dt.NewRow
        dr(0) = txtAccNo.Text
        dr(1) = txtTitle.Text
        dr(2) = txtSubtitle.Text
        dr(3) = cmbAuthorId.Text
        dr(4) = txtCallNo.Text
        dr(5) = txtEditor.Text
        dr(6) = txtCompilerId.Text
        dr(7) = txtTranslatorId.Text
        dr(8) = cmbMedium.Text
        dr(9) = txtISBN.Text
        dr(10) = txtPublisher.Text
        dr(11) = txtPuPlace.Text
        dr(12) = txtPubYear.Text
        dr(13) = txtEdition.Text
        dr(14) = txtNoPages.Text
        dr(15) = txtClassifiNo.Text
        dr(16) = txtSource.Text
        dr(17) = Val(txtPrice.Text)
        dr(18) = cmbBooktype.Text
        dr(19) = cmbCondition.Text
        dr(21) = dtpRegisteredDate.Value.Date
        dt.Rows.Add(dr)
        SqlDataAdapter1.Update(DataSet11, "Book")
    End If
End Class
```

```

        MsgBox("Successfully Saved!")
    End If
End Sub

Private Sub Clear()
    txtAccNo.Text = ""
    txtTitle.Text = ""
    txtSubtitle.Text = ""
    txtClassifiNo.Text = ""
    cmbAuthorId.Text = ""
    txtISBN.Text = ""
    txtPublisher.Text = ""
    txtEditor.Text = ""
    txtTranslator.Text = ""
    txtCompiler.Text = ""
    txtPuPlace.Text = ""
    txtPubYear.Text = ""
    txtEdition.Text = ""
    cmbBooktype.Text = ""
    txtNoPages.Text = ""
    txtCallNo.Text = ""
    txtPrice.Text = ""
    txtSource.Text = ""
End Sub

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
    Call Clear()
End Sub

Private Sub txtAuthorName_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtAuthorName.TextChanged
    Dim i As Integer
    For i = 0 To DataSet11.Tables("Author").Rows.Count - 1
        If cmbAuthorId.Text =
DataSet11.Tables("Author").Rows(i).Item(0) Then
            txtAuthorName.Text =
DataSet11.Tables("Author").Rows(i).Item(1)
        End If
    Next
End Sub

Private Sub Label23_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label23.Click
    NewAuthor.Show()
End Sub

Private Sub Label21_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label21.Click
    NewTranslator.Show()
End Sub

Private Sub Label22_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label22.Click
    NewCompiler.Show()
End Sub

Private Sub Label27_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label27.Click
    NewEditor.Show()
End Sub

```

## Coding for Add Members

```
Public Class Memre
    Dim dt As DataTable
    Dim dr As DataRow
    Private Sub Memre_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SqlDataAdapter1.Fill(DataSet11, "Member")
        dt = DataSet11.Tables("Member")
    End Sub
    Private Sub Clear()
        txtMemId.Text = ""
        txtAdNo.Text = ""
        txtFname.Text = ""
        txtAddress.Text = ""
        txtTele4n.Text = ""
    End Sub
    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnSave.Click
        If txtMemId.Text = "" Then
            MsgBox("Please enter the membership ID...")
            txtMemId.Focus()
        ElseIf txtFname.Text = "" Then
            MsgBox("Please enter the full name...")
            txtFname.Focus()
        ElseIf cmbMemType.Text = "" Then
            MsgBox("Please select member type...", , "LMS")
        ElseIf txtAddress.Text = "" Then
            MsgBox("Please enter your Address...", , "LMS")
            txtAddress.Focus()
        End If
        If MsgBox("Are you sure to save ?", MsgBoxStyle.YesNo, "Save") =
MsgBoxResult.Yes Then
            dr = dt.NewRow
            dr(0) = txtMemId.Text
            dr(1) = txtAdNo.Text
            dr(2) = txtFname.Text
            dr(3) = dtpJoindate.Value.Date
            dr(4) = Val(txtGrade.Text)
            dr(5) = Val(txtPrsentGr.Text)
            dr(6) = txtAddress.Text
            dr(7) = txtTele4n.Text
            dr(8) = cmbMemtype.Text
            If rdbBoarder.Checked = True Then
                dr(9) = "Boarder"
            ElseIf rdbDaySchooler.Checked = True Then
                dr(9) = "Dayschooler"
            End If
            dt.Rows.Add(dr)
            SqlDataAdapter1.Update(DataSet11, "Member")
            MsgBox("Saved")
        End If
    End Sub
    Private Sub cmbMemtype_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbMemType.SelectedIndexChanged
```

```

        If cmbMemType.Text = "Accadameic staff" Or cmbMemType.Text =
"Non Accadamic staff" Then
            rdbBoarder.Enabled = False
            rdbDaySchooler.Enabled = False
            txtAdNo.Enabled = False
            txtGrade.Enabled = False
            txtPrsentGr.Enabled = False
        Else
            rdbBoarder.Enabled = True
            rdbDaySchooler.Enabled = True
            txtAdNo.Enabled = True
            txtGrade.Enabled = True
            txtPrsentGr.Enabled = True
        End If
    End Sub
    'Telephone number validation
    Private Sub txtTele4n_Leave(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTele4n.Leave
        If Not IsNumeric(txtTele4n.Text) Then
            MsgBox("Please check your Phone number!",
MsgBoxStyle.Exclamation, "LMS")
            txtTele4n.Focus()
        ElseIf txtTele4n.TextLength <> 10 Then
            MsgBox("Please check your Phone number!",
MsgBoxStyle.Exclamation, "LMS")
            txtTele4n.Focus()
        ElseIf Mid(txtTele4n.Text, 1, 1) <> "0" Then
            MsgBox("Please check your Phone number!",
MsgBoxStyle.Exclamation, "LMS")
            txtTele4n.Focus()
        End If
    End Sub
    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
        Call Clear()
    End Sub
End Class

```



## Code for Lending

```
Public Class Lending
    Dim dt, dt1, dt2 As DataTable
    Dim dr, dr1 As DataRow
    Private Sub Lending_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SqlDataAdapter1.Fill(DataSet11, "Lending")
        dt = DataSet11.Tables("Lending")
        SqlDataAdapter2.Fill(DataSet11, "Book")
        dt1 = DataSet11.Tables("Book")
        SqlDataAdapter3.Fill(DataSet11, "Member")
        dt2 = DataSet11.Tables("Member")
        Dim today As System.DateTime
        today = System.DateTime.Now
        dtpTo.Value = today.AddDays(7)
    End Sub
    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
        dr = dt.NewRow
        dr(0) = txtLentId.Text
        dr(1) = txtMemID.Text
        dr(2) = txtMemName.Text
        dr(3) = txtAccNo.Text
        dr(4) = txtTitle.Text
        dr(5) = cmbBookCondition.Text
        dr(6) = dtpFrom.Value.Date
        dr(7) = dtpTo.Value.Date
        dt.Rows.Add(dr)
        SqlDataAdapter1.Update(DataSet11, "Lending")
        MsgBox("Saved")
        dr1 = dt1.Rows.Find(txtAccNo.Text) 'coding Update
        dr1.BeginEdit()
        dr1(20) = "Lent"
        dr1.EndEdit()
        SqlDataAdapter2.Update(DataSet11, "Book")
        Call Clear()
    End Sub
    Private Sub Clear()
        txtMemID.Text = ""
        txtAccNo.Text = ""
        txtTitle.Text = ""
    End Sub
    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
        Call Clear()
    End Sub
    Private Sub txtAccNo_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtAccNo.TextChanged
        Dim i As Integer 'Automatically fills the Title when entering
the accession number
        For i = 0 To dt1.Rows.Count - 1
            If txtAccNo.Text = dt1.Rows(i).Item(0) Then
                txtTitle.Text = dt1.Rows(i).Item(1)
            Exit For
        End If
    Next
End Class
```

```

End Sub

Private Sub txtMemID_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtMemID.TextChanged
    Dim i As Integer 'Automatically fill the Name when entering the
membership Id
    For i = 0 To dt2.Rows.Count - 1
        If txtMemID.Text = dt2.Rows(i).Item(0) Then
            txtMemName.Text = dt2.Rows(i).Item(2)
            Exit For
        End If
    Next
End Sub

Private Sub dtpFrom_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles dtpFrom.ValueChanged
    Dim today As System.DateTime
    today = System.DateTime.Now
    dtpTo.Value = today.AddDays(7)
End Sub
End Class

```

## Code for Search

```
Public Class Search
    Private Sub Search_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SqlDataAdapter1.Fill(DataSet11, "Member")
        SqlDataAdapter2.Fill(DataSet11, "Furniture")
        SqlDataAdapter3.Fill(DataSet11, "Book")
        SqlDataAdapter4.Fill(DataSet11, "Reservations")
    End Sub
    'Book Search
    Private Sub txtAccessionNo_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtAccessionNo.TextChanged
        If rdbAccessionNo.Checked = True Then
            With DataSet11.Tables("Book")
                With DataGrid1
                    .CaptionText = "Search by Accession NO"
                End With
                If txtAccessionNo.Text = "" Then
                    .DefaultView.RowFilter = "AccessionNo Like'%"
                Else
                    .DefaultView.RowFilter = "AccessionNo Like'" &
txtAccessionNo.Text & "%'"
                End If
                DataGrid3.DataSource = .DefaultView
            End With
        End If
    End Sub
    Private Sub cmbClassificationNo_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbClassificationNo.SelectedIndexChanged
        If rdbClassificationNo.Checked = True Then
            With DataSet11.Tables("Book")
                With DataGrid1
                    .CaptionText = "Search by Classification NO"
                End With
                If cmbClassificationNo.Text = "" Then
                    .DefaultView.RowFilter = "ClassificationNo Like'%"
                Else
                    .DefaultView.RowFilter = "ClassificationNo Like'" &
cmbClassificationNo.Text & "%'"
                End If
                DataGrid3.DataSource = .DefaultView
            End With
        End If
    End Sub
    Private Sub txtAuthor_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtAuthor.TextChanged
        If rdbAuthor.Checked = True Then
            With DataSet11.Tables("Book")
                With DataGrid1
                    .CaptionText = "Search by Author"
                End With
                If txtAuthor.Text = "" Then
                    .DefaultView.RowFilter = "Author Like'%"
                Else
                    .DefaultView.RowFilter = "Author Like'" &
txtAuthor.Text & "%'"
                End If
            End With
        End If
    End Sub
End Class
```

```

        .DefaultView.RowFilter = "Author Like'" &
txtAuthor.Text & "%'"
        End If
        DataGrid3.DataSource = .DefaultView
    End With
End If
End Sub
Private Sub txtCallNo_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtCallNo.TextChanged
    If rdbCallNo.Checked = True Then
        With DataSet11.Tables("Book")
            With DataGrid1
                .CaptionText = "Search by Call NO"
            End With
            If txtCallNo.Text = "" Then
                .DefaultView.RowFilter = "CallNo Like'%"
            Else
                .DefaultView.RowFilter = "CallNo Like'" &
txtCallNo.Text & "%'"
            End If
            DataGrid3.DataSource = .DefaultView
        End With
    End If
End Sub

Private Sub txtTitle_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtTitle.TextChanged
    If rdbTitle.Checked = True Then
        With DataSet11.Tables("Book")
            With DataGrid1
                .CaptionText = "Search by Title"
            End With
            If txtTitle.Text = "" Then
                .DefaultView.RowFilter = "Title Like'%"
            Else
                .DefaultView.RowFilter = "Title Like'" &
txtTitle.Text & "%'"
            End If
            DataGrid3.DataSource = .DefaultView
        End With
    End If
End Sub

'Member Search
Private Sub txtMembershipId_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtMembershipId.TextChanged
    If rdbMembershipID.Checked = True Then
        With DataSet11.Tables("Member")
            With DataGrid1
                .CaptionText = "Search by Membership ID"
            End With
            If txtMembershipId.Text = "" Then
                .DefaultView.RowFilter = "MembershipId like '%"
            Else
                .DefaultView.RowFilter = "MembershipId like '" &
txtMembershipId.Text & "%'"
            End If
        End With
    End If
End Sub

```

```

        DataGrid1.DataSource = .DefaultView
    End With
End If
End Sub

Private Sub txtAdmissionNo_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtAdmissionNo.TextChanged
    If rdbAdmissionNo.Checked = True Then
        With DataSet11.Tables("Member")
            With DataGrid1
                .CaptionText = "Search by Admission NO"
            End With
            If txtAdmissionNo.Text = "" Then
                .DefaultView.RowFilter = "AdmissionNo like'%"
            Else
                .DefaultView.RowFilter = "AdmissionNo like'" &
txtAdmissionNo.Text & "%'"
            End If
            DataGrid1.DataSource = .DefaultView
        End With
    End If
End Sub

Private Sub txtFullName_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtFullName.TextChanged
    If rdbFullName.Checked = True Then
        With DataSet11.Tables("Member")
            With DataGrid1
                .CaptionText = "Search by Full Name"
            End With
            If txtFullName.Text = "" Then
                .DefaultView.RowFilter = "Fname like'%"
            Else
                .DefaultView.RowFilter = "Fname like'" &
txtFullName.Text & "%'"
            End If
            DataGrid1.DataSource = .DefaultView
        End With
    End If
End Sub

'Member Search
Private Sub txtfurnitureId_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtfurnitureId.TextChanged
    If rdbFurnitureId.Checked = True Then
        With DataSet11.Tables("Furniture")
            With DataGrid1
                .CaptionText = "Search by Furniture ID"
            End With
            If txtFullName.Text = "" Then
                .DefaultView.RowFilter = "FurnitureId like'%"
            Else
                .DefaultView.RowFilter = "FurnitureId like'" &
txtfurnitureId.Text & "%'"
            End If
            DataGrid2.DataSource = .DefaultView
        End With
    End If
End Sub

```

```

        End If
    End Sub

    Private Sub txtFurniture_TextChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles txtFurniture.TextChanged
        If rdbFurniture.Checked = True Then
            With DataSet11.Tables("Furniture")
                With DataGrid1
                    .CaptionText = "Search by Furniture"
                End With
                If txtFullName.Text = "" Then
                    .DefaultView.RowFilter = "Furniture like'%"
                Else
                    .DefaultView.RowFilter = "Furniture like'" &
txtFurniture.Text & "%'"
                End If
                DataGrid2.DataSource = .DefaultView
            End With
        End If
    End Sub

'Reseervations Search
    Private Sub TextBox2_TextChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles txtReservationId.TextChanged
        If rdbReservationId.Checked = True Then
            With DataSet11.Tables("Reservations")
                With DataGrid1
                    .CaptionText = "Search by Reservation ID"
                End With
                If txtReservationId.Text = "" Then
                    .DefaultView.RowFilter = "ReservationId Like'%"
                Else
                    .DefaultView.RowFilter = "ReservationId'" &
txtReservationId.Text & "%'"
                End If
                DataGrid4.DataSource = .DefaultView
            End With
        End If
    End Sub

    Private Sub txtAccessionNoReserve_TextChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    txtAccessionNoReserve.TextChanged
        If rdbAccessionNoReserve.Checked = True Then
            With DataSet11.Tables("Reservations")
                With DataGrid1
                    .CaptionText = "Search by Accession No"
                End With
                If txtAccessionNoReserve.Text = "" Then
                    .DefaultView.RowFilter = "AccessionNo Like'%"
                Else
                    .DefaultView.RowFilter = "AccessionNo'" &
txtAccessionNoReserve.Text & "%'"
                End If
                DataGrid4.DataSource = .DefaultView
            End With
        End If
    End Sub

```

```

Private Sub txtMembershipIdReserve_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtMembershipIdReserve.TextChanged
    If rdbReservationId.Checked = True Then
        With DataSet11.Tables("Reservations")
            With DataGrid1
                .CaptionText = "Search by Membership ID"
            End With
            If txtReservationId.Text = "" Then
                .DefaultView.RowFilter = "MembershipId Like'%"
            Else
                .DefaultView.RowFilter = "MembershipId'" &
txtMembershipIdReserve.Text & "%'"
            End If
            DataGrid4.DataSource = .DefaultView
        End With
    End If
End Sub
End Class

```

## Code for fine calculation

```
Public Class Fine
    Dim dt, dt1 As DataTable
    Dim dr As DataRow

    Function autoNo() As String
        If DataSet11.Tables("Fine").Rows.Count = 0 Then
            Return "FI00001"
        Else
            Dim i As Integer
            i = DataSet11.Tables("Fine").Rows.Count - 1
            Return
                Format(Val(Mid(DataSet11.Tables("Fine").Rows(i).Item(0), 3)) + 1,
                    "\FI00000")
        End If
    End Function

    Private Sub Fine_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SqlDataAdapter1.Fill(DataSet11, "Fine")
        SqlDataAdapter2.Fill(DataSet11, "Lent")
        dt = DataSet11.Tables("Fine")
        txtfineId.Text = autoNo()

        Dim x As Double 'fine calculation
        x = Val(txtNoOfDyLate.Text)
        Label10.Text = x * 10.0

    End Sub

    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
        dr = dt.NewRow
        dr(0) = txtfineId.Text
        dr(1) = txtlentId.Text
        dr(2) = txtMemId.Text
        dr(3) = Label10.Text
        dr(4) = dtpFinedate.Value
        dr(5) = txtDescription.Text
        dt.Rows.Add(dr)
        SqlDataAdapter1.Update(DataSet11, "Fine")
        MsgBox("Successfully Saved!")
    End Sub

    Private Sub cmbFinefor_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbFinefor.SelectedIndexChanged
        txtDescription.Text = cmbFinefor.Text
    End Sub

End Class
```



# Appendix G - Client Certificate

## ST. JOHN'S COLLEGE, JAFFNA SRI LANKA.

*Principal:*

**The Revd. N. J. Gnanaponrajah,**

B.A. (Sri Lanka),  
Te. Ce. Pr. Ed (Sri Lanka),  
P.G.D.R.E. (Birmingham, UK),  
P.G.D.E.M. (Sri Lanka),  
C.SASCP. (Florida, USA).

Tel/Fax : (0094) 021-222 2432  
E-mail : stjohns@sltnet.lk / sjc.jaffna@gmail.com  
Web : www.sjc.jaffna.com

(Founded by CMS, 1823)



*Vice Principals:*

**V. S. B. Thuseetharan,**

B.Sc. (Sri Lanka),  
P.G.D.E (Sri Lanka),  
M.Ed. (Sri Lanka).

**A. H. Gnanarajan,**

B.A. (Jaffna), M.A. (P.Admin) (Jaffna),  
Sp. Trained - English, P.G.D.E. (Jaffna),  
TKT (Cambridge), English F.L. A. (Maryland)

"LUX IN TENEBRIS LUCET"

LIGHT SHINETH IN DARKNESS \* இருளில் ஒளி பிரகாசிக்கிறது \* අලංකය අන්ධකාරයේ දීදලයි

3.10.2017

Coordinator,  
BIT Degree,  
University of Colombo School of Computing

Respected Sir/Madam,

### Library Management System for St. Johns College Jaffna

This is to bring to your kind notice that Mr. J .R .Daniel Nitharshan (893630112V) has completed the installation of the above mentioned system into the computer of our school library.

Thank You

Yours truly,

**The Revd. N. J. GNANAPONRAJAH**  
Principal  
St. John's College  
Jaffna, Sri Lanka.



SCOUTS CENTENARY 2016 - "SUCCESS THROUGH SCOUTING"

# **GLOSSARY**

Class diagram – shows the object classes of the system and the relationships between

Diagram – A diagram used to show the flow of a business process

Interviews – A fact finding technique whereby the systems analyst collects information from individual through face to face interaction [whitten,2007]

Observation – A fact finding technique wherein the systems analyst either participates in or watches a person perform activities to learn about the system [whitten,2007]

Use Case Diagram – A use case diagram displays the relationship among actors and use cases.

Use Case – A business event for which the system must provide a defined response  
Activity