



|  |  |  |
| --- | --- | --- |
| Subject Title | : | Principles and Practices of Software Production |
| System Name | : | Computerized Library Management System |
| Intake Title | : | Computing |
| Intake Code | :P | PT1082 |
| Assignment TTTTTTTTTytTypeType | : | Group Assignment |
| Hand-Out-Date | : | 16th August, 2012 |
| Hand-In-Date | : | 27th October, 2012 |

**Submitted To:**

**Group No:** 6xxxxxxxxxxx

**Group Members:** **(Module Lecturer)**

xxxxxxxx (PT1082214)

Bhagyashri B (PT1082238)

xxxxxxxxxxx (PT1082227)

Table of Contents

[1.0 ACKNOWLEDGEMENT 5](#_Toc339066437)

[2.0. ABSTRACT 6](#_Toc339066438)

[3.1. Organization 7](#_Toc339066439)

[3.1.1 Necessity of LMS 7](#_Toc339066440)

[3.2. Task 7](#_Toc339066441)

[3.3.1. Scope of the proposed system 7](#_Toc339066442)

[3.3.2. Scope of the user 7](#_Toc339066443)

[3.4. Objectives of the proposed system 7](#_Toc339066444)

[4.0. PROBLEM BACKGROUND 8](#_Toc339066445)

[4.1. Working of existing system 8](#_Toc339066446)

[4.2. Existing problems 8](#_Toc339066447)

[4.3. Assumptions 9](#_Toc339066448)

[5.0. PROPOSED SOLUTION 9](#_Toc339066449)

[5.1. Solution of existing problems 9](#_Toc339066450)

[5.2. Additional Features 10](#_Toc339066451)

[5.3. Modules 10](#_Toc339066452)

[6.0. PROJECT PLANNING CONTROL 10](#_Toc339066453)

[6.1. Work Breakdown Structure (WBS) 10](#_Toc339066454)

[6.3. Staffing plan 11](#_Toc339066455)

[7.0. PROJECT MANAGEMENT 13](#_Toc339066456)

[7.0. PROJECT MANAGEMENT 13](#_Toc339066457)

[7.1. Introduction 13](#_Toc339066458)

[7.2. Distinctive characteristics 13](#_Toc339066460)

[7.3. Tasks undertaken by team 13](#_Toc339066461)

[7.4. Project Planning 14](#_Toc339066462)

[7.5. Project Planning Process related to LMS 14](#_Toc339066463)

[7.6 Resource allocation and role 15](#_Toc339066464)

[8.0. SCHEDULE PLANNING 15](#_Toc339066465)

[8.1. Definition and importance 15](#_Toc339066466)

[8.2. Milestones and deliverables 15](#_Toc339066467)

[8.3.1. Tool used 17](#_Toc339066468)

[9.0. SELECTION OF METHODOLOGY 18](#_Toc339066469)

[9.1 Introduction 18](#_Toc339066470)

[9.2 Purpose and scope 18](#_Toc339066471)

[9.3. Justification 18](#_Toc339066472)

[9.4. Features/characteristics 18](#_Toc339066473)

[10.0. COST ESTIMATION 20](#_Toc339066474)

[11.0. RISK MANAGEMENT 23](#_Toc339066476)

[11.1. Introduction 23](#_Toc339066477)

[11.2. Basic approach of risk management 24](#_Toc339066478)

[11.3. Risk Assessment 24](#_Toc339066479)

[11.2.1. Risk Identification 24](#_Toc339066480)

[11.2.2. Risk Analysis 25](#_Toc339066481)

[11.2.3. Assumptions 25](#_Toc339066482)

[12.0. REQUIREMENT ENGINEERING 26](#_Toc339066483)

[13.0 Process Modelling 33](#_Toc339066484)

[13.1. Introduction 33](#_Toc339066485)

[13.2. Logical Data Modelling 33](#_Toc339066486)

[13.2.1. Context Diagram 33](#_Toc339066488)

[14.0 Data Modelling 39](#_Toc339066489)

[15.0 Data Dictionary for Proposed Systems 41](#_Toc339066490)

[16.0. DESIGN PRINCIPLES AND CONCEPTS 53](#_Toc339066491)

[17.0. ARCHITECTURAL DESIGN 57](#_Toc339066492)

[18.0. INTERACTIVE SCREEN DESIGNS 58](#_Toc339066493)

[19.0. PROGRAMMING ENVIRONMENT 63](#_Toc339066494)

[20.0. TESTING 63](#_Toc339066495)

APPENDICES………………………………………………………………………………..65

REFERENCES……………………………………………………………………………….68

# 1.0 ACKNOWLEDGEMENT

The development team takes this privilege to acknowledge our respected project supervisor**, Mrs. xxxxxxx**, for her timely guidance and tireless efforts that she provided us to help gain a great learning experience. Her source of inspiration and motivation resumed throughout the design, implementation and evaluation of the project. The topics of the lectures were extremely helpful in documenting the project.

Secondly, the team appreciates the work of those (colleagues) who indirectly helped them in suggesting valuable requirements and providing constructive criticism that benefited a lot in delivering the system. The vast internet resources provided much information regarding the project.

Thirdly, we gained a lot from the vast collection of books of APIIT library. It is the team’s utmost pleasure to express warm thanks for the encouragement, co-operation and support of those whose name has not appeared here but should not go unnoticed for helping the team members to successfully complete the project.

Last but not the least; the team is grateful to the Almighty for keeping his grace upon the team members during the whole development phase of the project.

xxxxxxxx

BHAGYASHRI

xxxxxxxxx

# 2.0. ABSTRACT

The project involves developing a “**Computerized Library Management System**” which includes how to implement various functionalities of a library to give information about the library.

For achieving the requirements, the objectives and deliverables were taken into consideration. The work was divided among the team members to plan, analyze the existing system and to fully implement the all the requirements of the proposed system. Problem background of the existing system is studied in detail and an appropriate solution is suggested. The problems faced and how they were managed during the development of the system is also discussed. Testing was performed to remove any errors by using proper technique and logic.

The basic operations which are to be performed using this system are: registering new members and books, updating, editing and viewing information of books and students, searching for books and members, transaction (issue and return) of books, maintain late fine of the students who return books after due date, maintain records, check how many books are issued to whom and stock available by generating reports.

The development team decided to use Visual studio 2010 with visual basic language as front end and sql server 2008 as back end.

**3.0. INTRODUCTION**

## 3.1. Organization

The Library Management System i.e. LMS which is to be developed is suggested by the principal of **XXX**. It has a small library (no online transactions like purchase) which does daily activities of a library like record keeping of books and students, check in and checkout of books, calculating fine etc. So, the institute wants to switch over from regular handwritten method to computerized one.

### 3.1.1 Necessity of LMS

When records are maintained manually, there is a greater risk of incomplete and ambiguous information. At the time of rush, it is difficult to access the registers at a time. Books may be lost, students cannot view information about books by themselves, defaulters could not be found easily, and fine calculation would become error prone etc.

## 3.2. Task

The main purpose of the application to be developed is to provide a system which can be easily accessed, to improve library services and reduce paperwork. It concentrates on basic operations in a library**.**

**3.2.1. Target audience**

The users of the proposed system are mainly the librarian who has the highest access to make any changes and the students who can only view the details of the books available.

**3.3. Scope**

### 3.3.1. Scope of the proposed system

The application can be used by any College Library as well as school library. Since it is not an online system, books cannot be purchased. It cannot be used in central library to be accessed by other departments.

### 3.3.2. Scope of the user

Stakeholders are persons who are directly or indirectly affected by the success or failure of the system. **Stakeholders** involved are librarian, students. Students can only view the details of the list of books available but the librarian can search, edit, update, edit, delete or add any new member (books or students) as required. The students cannot issue books themselves.

## 3.4. Objectives of the proposed system

The application will automate the existing system of manually maintaining the records .Its purpose is to provide a multi-user version and can take care of all the fundamental functions of a Library like Cataloguing, Circulation, and Accessioning. Some of the basic objectives are:

* **Save time and resources and operate easily:** when records are manually maintained, it requires much time and space as well as manpower to maintain it efficiently. User can retrieve information in a very less time since the user-interface to be designed will be user-friendly and usable.
* **To exclude paperwork:** In order to make the system more economical, the use of new system does not need registers and files that create confusion and errors.
* **To increase accuracy, efficiency and reliability:** The output available from the system will be accurate, as expected and with no errors and chaos. The proposed system will be reliable since now system will give complete information any time with accuracy.

# 4.0. PROBLEM BACKGROUND

## 4.1. Working of existing system

The staffs perform **labour-intensive** tasks. Whenever a new student comes, his details are recorded in a register. To search for any detail, lots of pages of registers are turned. To modify or edit any information, overwriting and cutting is done. To delete the details, pages are torn off. To calculate fine, days are counted manually.

## 4.2. Existing problems

Library staff faces a lot of problems in maintain the records. They are:

* **Time-consuming and inaccurate (Error prone), Unreliable:** Since the records are maintained manually, it takes much time to operate information of students or books. There is a chance that the information obtained may be inaccurate. There may be errors in fine. It is very difficult to find the defaulters who have not returned the book for long period of time.
* **Problems in record maintenance (inflexible):** Records of books and students need to be maintained from the time of new entry till its deletion or removal. It is very difficult to keep track of: which books are issued to whom, when the book is to be returned, what is the fine to be imposed etc
* **Lack of immediate retrievals:** The information is very hard to retrieve and to find particular information like- E.g. - To find out about a student’s history, the user has to go through various registers. He needs to search through unnecessary details. This results in **inconvenience** and wastage of time.

**Problems in report generation:** When records are maintained manually, reports cannot be generated.Therefore, if questions like how many books are issued, the number of books of a particular subject etc. in a library remains unanswered.

* **Insecure, inefficient and unproductive:** There is a chance of **loss** or damage of records. No security or privacy is there for protection of information.

## 4.3. Assumptions

It is assumed that presently the organization **maintains manual records** and not a computerized one. The development team has studied the problems in the existing system and will develop a computerized LMS to resolve the problems faced.

# 5.0. PROPOSED SOLUTION

## 5.1. Solution of existing problems

* **Quick retrieval of information** related to books or students since user can login and search by id or book number or code. Since the interface designed is highly interactive, user can easily access it.
* **Less time-consuming and accurate:** Calculator is provided in the system so it will take very less time to calculate fine. No data redundancy since now proper techniques is used to store data in database. No paper work hence space as well as much time can be saved and less error prone.
* **Automated report generation:** Reports will be generated automatically by the system without searching here and there into registers. The information about number of books available etc. can easily be traced out.
* **Secure, productive and efficient:** Strong security is provided and only authenticated users can use the system by entering username and password. Proper error messages are prompted in exclamation and dialog boxes thus system warns the user of any error by giving proper feedback.
* **Records easily maintained:** New students and books can be easily added to the database. There will be no loss of books or information since the system will keep a track of date, issued, to whom a book is issued, when it should be returned etc. Every member and books have a unique ID by which they can be sorted out easily.

Taking into consideration the problems faced in the use of registers, a new system is proposed with the following features:

## 5.2. Additional Features

* Students can also view the number of books available in the library and allows them to search for a particular book.
* **Convenient**, user friendly interface- more usable, look and feel Environment.
* Spot registrations can be performed.
* Search, edit, view, update, add and delete facility- It provides a guest login for the members to view details about the available books and allows them to search for a particular book. Quick transaction or fast processing-issue and return of books and fine calculation fines made easier.
* Vast amount of information available related to any student or book.

## 5.3. Modules

The Library Management system will include the following **modules**:

* **Login module**: (interactive user interface).Each member has a user id and password to login into the system as well provides security
* **Librarian:** Librarian has the greatest access to the system. He/she can modify, edit, update, delete, search or add students or books.
* **Students:** Students have the least priority only to view the list of books available. A student is also given a username and password.
* **Books:** Books can be circulated (issue and return of books). It can be found out how many books are available and the students to whom it is allotted.
* **Fine calculation:** Fine is calculated by the librarian if a student returns a book after the due date (10 Rs per day).

# 6.0. PROJECT PLANNING CONTROL

Project planning discussed in next section (project management).

## 6.1. Work Breakdown Structure (WBS)

A work breakdown structure (WBS) is a chart in which the critical work elements, called tasks, of a project are illustrated to portray their relationships to each other and to the project as a whole. It starts with the major project areas to be accomplished and breaks into actionable pieces of work, segmenting elements into appropriate sublevels.

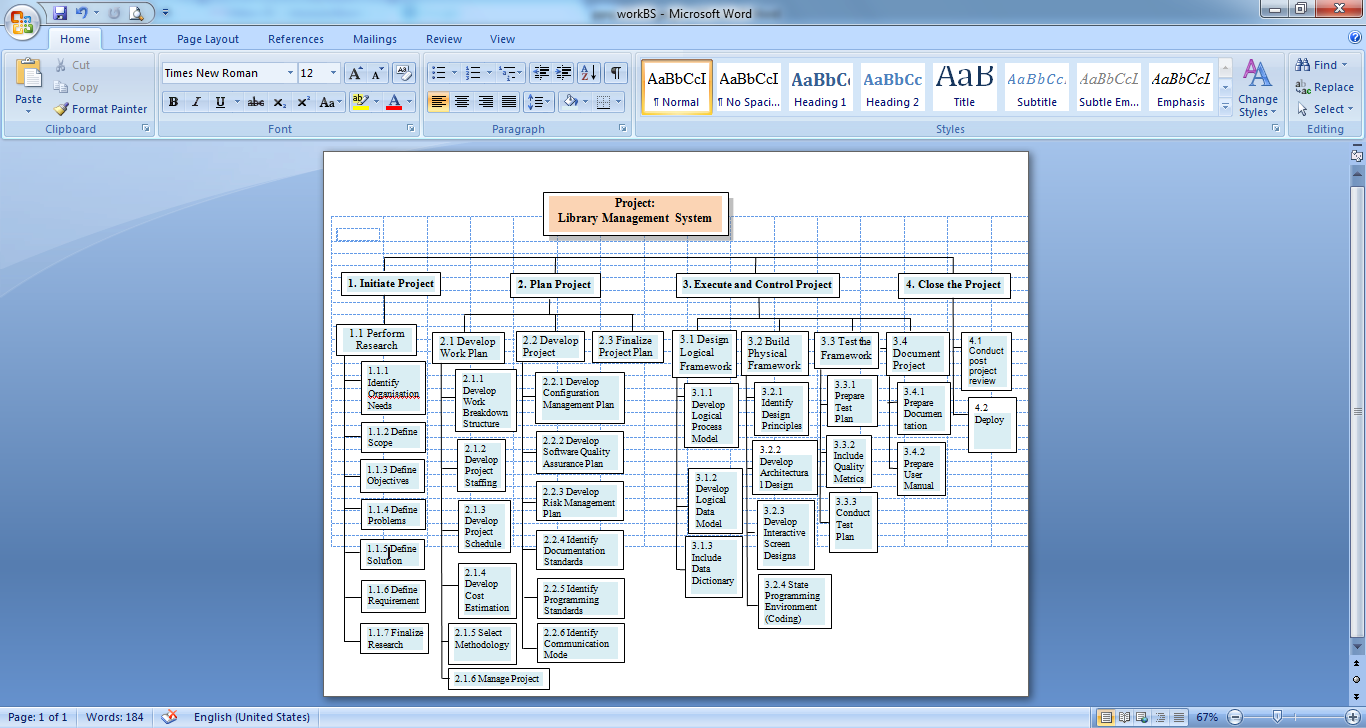
**6.2. Project Organization Chart**

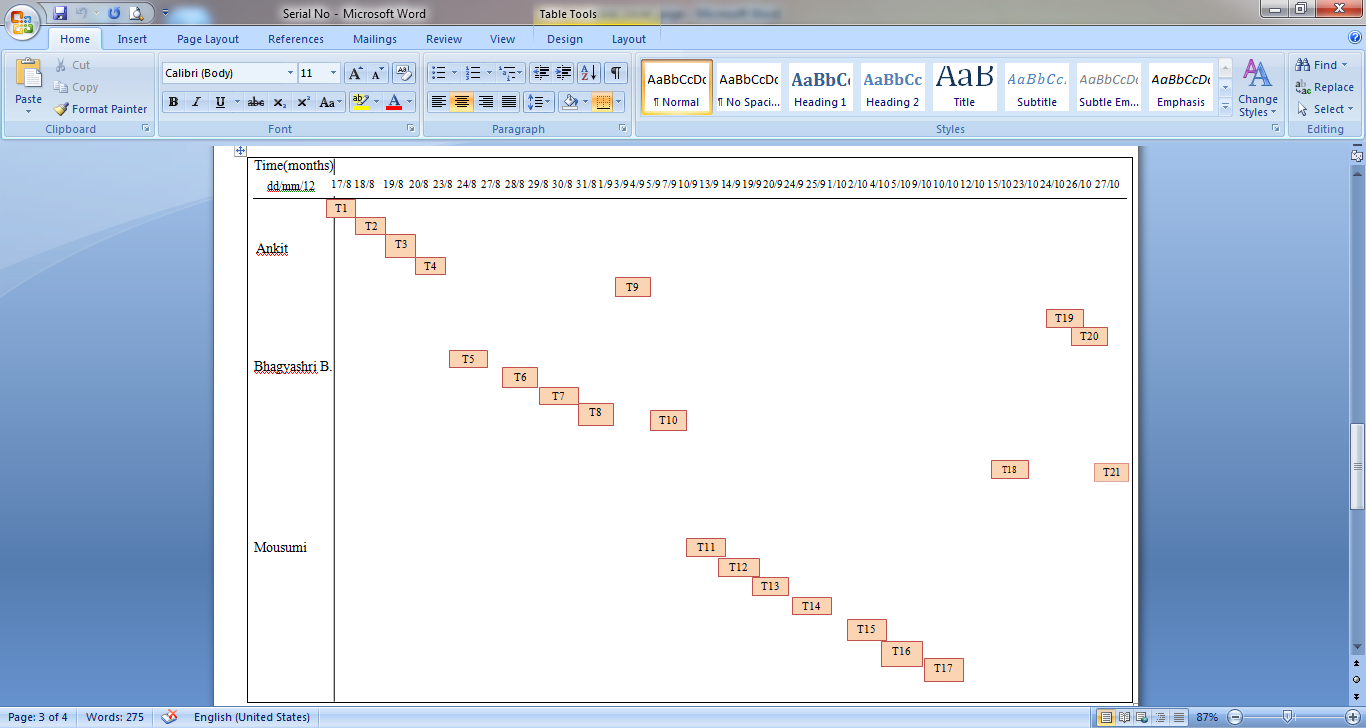
Figure 4: Work Breakdown structure

It is a graphically detailed document that outlines the particular and specific project team members that have been assigned to work on a particular project, as well as how exactly these individual project team members will interrelate with one another over the course of the project life cycle as well as defining crystal clear their specific responsibilities and ‘ways of working' for the team.

Figure 5: Project organization chart

## 6.3. Staffing plan

The purpose of the staffing plan is to make certain that the project has sufficient staff with the right skills and experience to ensure a successful project completion. It helps organizations better plan for the future in identifying current and future staffing needs.



**6.4. Other plans**

**Configuration management plan:** Using CM plan, software development team outlines how they will control and monitor changes to their code base, and how to merge those changes together without destroying the functionality of the code. In future, LMS can have many more functionalities since it can accommodate change.

**Software quality assurance plan:** It is the establishment of a framework of organizational procedures and standards that lead to high-quality software. **(Refer to appendix B)**

**Risk management plan:** **(Refer to appendix A)**

**6.5. Documentation standards**

|  |  |  |
| --- | --- | --- |
| SL. No. | Properties considered | Standards |
| 1. | Heading size | 16 points |
| 2. | Sub heading size | 14 points |
| 3. | Size of words in paragraphs | 12 points |
| 4. | Line spacing between lines and paragraphs | 1.5 |
| 5. | Font style | Times New Roman |
| 6. | Indentation | Tab space in starting a paragraph |
| 7. | Alignment | Justified |

**6.6. Programming standards**

|  |  |  |
| --- | --- | --- |
| SL. No. | Features considered | Standards |
| 1. | Goto statement | No use of goto statement made |
| 2. | Comments | Comments have been provided. |
| 3. | Procedures and functions | Functions are used to prevent repetition of code. |
| 4. | Indentation | Properly indented |
| 5. | Frond end language used | Visual basic |
| 6. | Back end language used | SQL |
| 7. | Naming of variables | Self-describing and camel casing used |

**6.7. Communication mode**

**Notations used**

Written  **- - - - - - - ->** Verbal

**Designations**

1- Project manager

2- Analyzer 4- Coder/developer

3- Designer 5- End user/customer

# 7.0. PROJECT MANAGEMENT

# 7.0. PROJECT MANAGEMENT

## 7.1. Introduction

## Software project management is concerned with activities involved in ensuring that software is delivered on time and in accordance with the requirements of the organizations developing and procuring the software. It includes project planning, project scheduling and risk management. It is a process of scoping, planning, estimating, scheduling, staffing, organizing, directing & controlling and closing the development of an acceptable system.

## 7.2. Distinctive characteristics

* Assignment of integrative responsibilities related to each project, program and project portfolio.
* Application of integrative and predictive practices, methods, systems and tools for producing and effectively using the information required to plan, schedule, monitor, and control the scope, risks, schedules, resources and costs of projects
* Building and directing each project and program team

## 7.3. Tasks undertaken by team

* **Proposal writing:** write a proposal which consists of objectives of the project, how it should be carried out, cost and schedule and estimates. Since LMS is a student project, no proposal writing was needed, project objective was known from before.
* **Project planning and scheduling and cost:** Identifying the activities, milestones and deliverables produced by a project and estimating the resources (cost estimation). For LMS, project scheduling and cost estimation is discussed in next section.
* **Project monitoring and reviews:** The manager keeps of the project’s progress and compares actual and planned progress and costs. For e.g. in LMS, the project was reviewed regularly by the module lecturer and the members constantly monitored each activity of our system in terms of cost, time and resource usage to minimize error.
* **Personnel selection and evaluation:** Select skilled staff with appropriate experience. For e.g. in LMS, the group members together developed the project, the group was form
* **Report writing and presentations:** Write concise, coherent documents that abstract critical information from detailed project For LMS, a document and slides were made to present the project.

# 7.4. Project Planning

It is a continuous activity from initial concept through to system delivery. Plans must be regularly revised as new information becomes available. In the planning phase, the exact parameters of the project are defined and ensure that all the pre-requisites for Project Execution and Control are done properly. It sets out resources available, work breakdown and schedule for work. For LMS, resource allocation chart, WBS and Gantt chart have been prepared.

# 7.5. Project Planning Process related to LMS

Establish the project constraints (set out LMS’s goals)

Make initial assessments of the project parameters (Find out resources, cost etc required)

Define project milestones and deliverables (Gantt chart, pert chart, WBS etc)

While project has not been completed or cancelled loop

Draw up project schedule

Initiate activities according to schedule

Wait (for a while)

Review project progress

Revise estimates of project parameters

Update the project schedule

Re-negotiate project constraints and deliverables

If (problems arise) then

Initiate technical review and possible revision

End if

End loop

# 7.6 Resource allocation and role

It consists of description of the resource, a statement of availability, time when the resource will be required and duration of time that resource will be applied. The resources needed for LMS are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Resource | Description (Role) | Available | Allocated |
| Manpower | It is the number of people necessary for completing the project. They may be experienced or skilled. | 3 | 3 |
| Material | The hardware, software and other resources necessary. | Visual studio 2008, 2010, SQL database, access database | Made use of VS 2010 and SQL database. |
| Service | Any internet facility or feedback provided. | 24 hours internet facility available with 14 GB pack. | Made use of 7GB |
| Time | The estimated time within which the project has to be completed. | 3 months  Each phase nearly took 15 to 20 days. | 3 months |
| Money | Total budget or cost estimate required | 4,00,000 Rs | Rs 2,05,020 |
| Location | The place where the development was carried out. | No particular cabin or workplace. | Since it is a student project, made by own. |

# 8.0. SCHEDULE PLANNING

## 8.1. Definition and importance

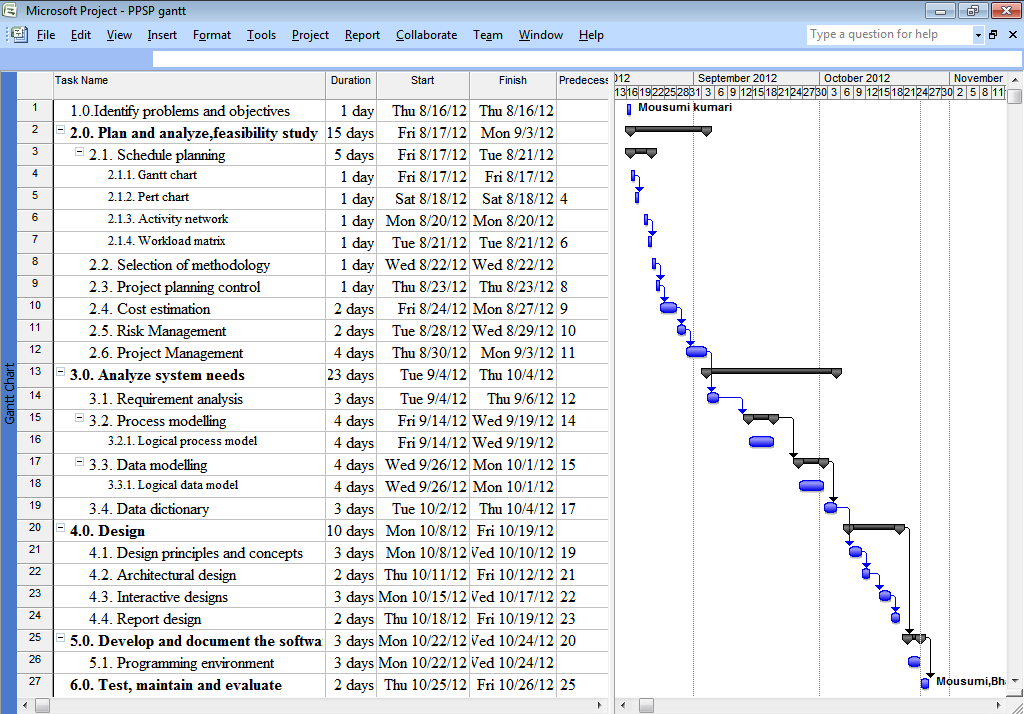
It is used by the project manager to commit people to the project and show the organization how the work will be performed. Identifying the tasks that need to be carried out and their dependencies and organizing them into a sequence of, estimate resources for activities , estimating how long they will take; allocating resources (mainly personnel) and scheduling when the tasks will occur.

## 8.2. Milestones and deliverables

A **milestone** is a recognizable end-point of a software process activity. It should be a former output. A **deliverable** is a project result that is delivered to the customer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Serial No. | Task No. | Task Name | Duration (MM/DD/YYYY)    **Start**  **(**  **Finish** | Milestones/Deliverables |
| 1. | T1 | Introduction | Fri 8/17/12 Fri 8/17/12 | - |
| 2. | T2 | Problem Background | Sat 8/18/12 Sat 8/18/12 | - |
| 3. | T3 | Proposed Solution | Sun 8/19/12 Sun 8/19/12 | - |
| 4. | T4 | Schedule Planning | Mon 8/20/12 Thu8/23/12 | Gantt chart, pert chart(M) |
| 5. | T5 | Selection of Methodology | Fri 8/24/12 Fri 8/24/12 | - |
| 6. | T6 | Project Planning Control | Mon 8/27/12 Tue 8/28/12 | Project Organization Chart(M) |
| 7. | T7 | Cost Estimation | Wed 8/29/12 Thu 8/30/12 | - |
| 8. | T8 | Risk Management | Fri 8/31/12 Sat 9/1/12 | - |
| 9. | T9 | Project Management | Mon 9/3/12 Mon 9/4/12 | - |
| 10. | T10 | Requirement Analysis | Mon 9/5/12 Mon 9/7/12 | SRS (M) |
| 11. | T11 | Process Modeling | Mon 9/10/12 Thu 9/13/12 | DFD(M) |
| 12. | T12 | Data Modeling | Fri 9/14/12 Wed 9/19/12 | ERD,ELH(M) |
| 13. | T13 | Data Dictionary | Thu 9/20/12 Mon 9/24/12 | - |
| 14. | T14 | Design Principles and Concepts | Mon 9/25/12 Mon 10/1/12 | - |
| 15. | T15 | Architectural Design | Tue 10/2/12 Thu 10/4/12 | - |
| 16. | T16 | Interactive Screen Design | Fri 10/5/12 Tue 10/9/12 | Interfaces System design documents (M) |
| 17. | T17 | Report Design | Wed 10/10/12 Fri 10/12/12 | Storyboarding, Interface design document   (M) |
| 18. | T18 | Programme Environment | Mon 10/15/12 Tue 10/23/12 | - |
| 19. | T19 | Testing | Wed 10/24/12 Fri 10/26/12 | Unit test plan, System test plan |
| 20. | T20 | Overall Documentation | Fri 10/26/12 Fri 10/26/12 | User Manual(D) |
| 21. | T21 | Deployment | Fri 10/27/12 Fri 10/27/12 | Software(D) |

**8.3. Gantt chart**

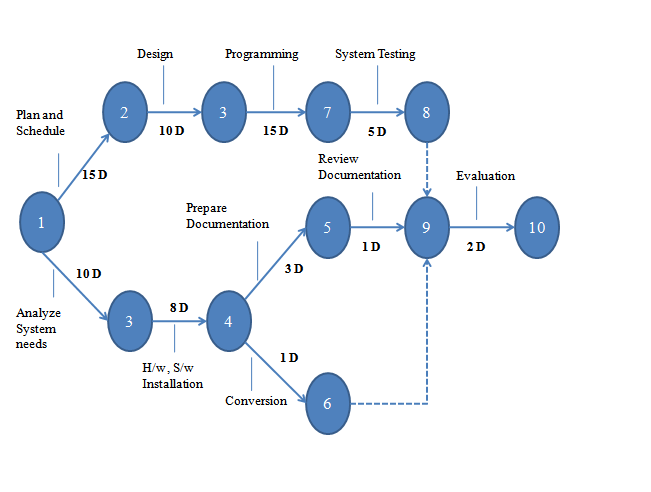
A Gantt chart is a graphical representation of the duration of tasks against the progression of time. It is useful tool for planning and scheduling projects and helpful when monitoring a project's progress. 

### 8.3.1. Tool used

The tool used by the development team to create Gantt chart is **Microsoft project 2007**. It allows people to create and manage project, this is done week by week or day by day depending on how you set it out and how you want it to be displayed.

**8.4. PERT chart**

A Program Evaluation Review Technique chart is a project management tool used to schedule, organize, and coordinate tasks within a project. It is a graphic representation of a project’s schedule, showing the sequence of tasks, which tasks can be performed simultaneously, and the critical path of tasks that must be completed on time in order for the project to meet its completion deadline.



# 9.0. SELECTION OF METHODOLOGY

## 9.1 Introduction

The methodology selected for developing **Library Management System** is **W**aterfall **M**odel. It is also called the classic life cycle, which suggests a systematic, sequential approach to software development that follows various phases.

## 9.2 Purpose and scope

The purpose of waterfall model is to plan how the team will approach towards the development of the given system. It also states how the tasks such as research, analysis, requirement gathering, project scheduling, cost estimation, logical and physical designing, testing and debugging, deployment will be carried out in systematic and sequential manner so that each next task is initiated after the completion of the preceding one.

## 9.3. Justification

The development team has chosen waterfall model for software development due to the following reasons:

* Waterfall model is meant for small or student projects. Since, LMS is a small project; waterfall model can be used to develop it. It is simple to implement and minimal resources are required to implement it.
* Moreover, the requirements are very clear and stable. So, there would be fewer amounts of risk and uncertainty.
* The project requires the fulfillment of one phase, before proceeding to the next. Therefore if there is a fault, it will be detected during one of the initial phases and will be sealed off for correction.
* Documentation is produced at every stage of the software's development. This makes understanding the product designing procedure, simpler.
* It clearly divides the problem into distinct phases that may be performed independently.

## 9.4. Features/characteristics

* The project consists of sequential, non-overlapping phases, where a phase cannot begin until the previous phase is already completed.
* Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance.
* At the end of every phase, there is a gate where a decision is made to allow the project to move forward or not. The product is only finished at the end of the last phase.
* At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project.
* Low overhead of the methodology compared to other methodologies (such as Agile).

**9.5. Structure/Steps**

The structure of the waterfall model is represented as follows:

**Planning**

Estimating

Scheduling

Tracking

**Communication**

Project initiation,

requirements gathering

**Modeling**

Analysis

Design

**Deployment**

Delivery

Support

Feedback

**Construction**

Code

Test

* **Communication:** The development team performed secondary research i.e. gone through earlier Library Management System so that the basic functioning and existing problems of it are understood. A meeting was organized by project manager between analysts, developer and end users to gather information about the system to be developed and the existing working criteria. Several questions were asked to the users of the system by consulting them like

**Why?** Why does user need this application/process/procedure?

Why do customers this method? Why is this problem?

**What?** What need/ constraints does this application/step satisfy?

**Where?** Where is this application utilized?

**Who?** Who uses this application?

**When?** When is it done? When does it have to be done?

**How?** How is it accomplished?

Finally, a **Requirement Specification document,** top-level analysis and design were documented to serve the purpose of guideline for the next phase of the model.

* **Planning:** When requirements were gathered, the team made an estimation of how LMS should function. Then, system design was done i.e. functionality of hardware and software was separated-out. LMS to be developed was assessed for quality and a test plan was prepared. A time schedule was chalked out to find out how the activities are to be completed and by when. Thus, a **system design specification** was prepared to see how LMS would look like in real.
* **Modeling:** After the confirmation of the system design by the organization for which the LMS is developed, algorithm was developed. The team decided to use SQL server 2008 as database and visual studio 2010.Here the actual work begins. Every type of resource which will be required for the smooth designing of the software was mentioned here .The algorithm of the process in which the software needs to be designed was made in this phase.
* **Construction:** Then algorithm was converted to small units of programs which were later integrated. Each unit was tested. For e.g., at first code for login module was tested followed by code of issue and return of books and do on. Then, integration testing was done to test the LMS as a whole. During testing, errors and flaws were removed. For e.g., lack of proper validation would cause the user to enter wrong input.
* **Deployment:** After development of LMS, it was installed in the system of the library. Old files were converted to a new one. Librarian was instructed how to use the software. Not much problems were found.. Any further changes could also be made if needed.

# 10.0. COST ESTIMATION

Cost estimation is used for generating request for proposals, contract negotiations, scheduling, monitoring and control. These estimates are necessary to establish a budget for the project or to set a price for the software for the customer. It can be used to determine what resources to commit to the project and how well these resources will be used.

**10.1. Techniques of cost estimation**

* **Algorithmic cost modelling**: In this technique cost is estimated as a mathematical function of product, project and process attributes whose values are estimated by project managers.
* **Expert judgement:** One or more experts in both software development and the application domain use their experience to predict software costs. Process iterates until some consensus is reached
* **Estimation by analogy:** The process of finding one or more projects that is similar to the one to be estimated in the same application domain.
* **Parkinson's Law:** The software cost is estimated to be whatever the customer has available to spend on the project. The estimated effort depends on the customer’s budget and not on the software functionality.
* **Pricing to win:** The software cost is estimated to be whatever the customer has available to spend on the project. The estimated effort depends on the customer’s budget and not on the software functionality.

**10.2. Technique used in cost estimation of LMS**

**COCOMO (Cost Constructive) Model 2**

It is an empirical, well-documented, ‘independent’ model which is not tied to a specific software vendor. It was derived by collecting data from a large number of software projects, then analyzing that data to discover formulae that were best-fit to the observations.

It has 4 different models: The Application Composition Model, The Early Design Model, The Post-Architecture Model and Reuse Model.

**Effort calculation**: The effort is calculated using formula of application-composition model which makes use of application points.At an early development stage, it is difficult to estimate the number of lines of code in a system.Thus, it is hard to estimate FP. In **LMS object point count is used** because of the following reasons, they are easier to estimate from a specification than function points as they are simply concerned with screens, reports and programming language modules.They can therefore be estimated at a fairly early point in the development process. It is calculated as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Complexity | Number | Complexity \*Number |
| No. of screens | 1 | 7 | 7 |
|  | 2 | 5 | 10 |
| 3 | 0 | 0 |
| No. of reports | 2 | 2 | 4 |
| 5 | 0 | 0 |
|  | 8 | 0 | 0 |
| No of modules | 10 | 14 | 140 |

|  |
| --- |
| Total OPC= 157 |

Final formula for effort computation is

**PM = (NAP\*(1-(%reuse/100)))/PROD** where NAP is the total number of application points in the delivered system. % reuse is the estimate of the amount of reused code in the development and PROD is the object-point productivity. Since, the development team has 10% of reuse component,

Effort = (NAP\*(1-0.1))/PROD [NAP=OPC=157]

= (157\*0.9)/PROD= 141.3/PROD

For calculating PROD, using values from the object point productivity table,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Developer’s experience and productivity.** | **Very low** | **Low** | **Nominal** | **High** | **Very High** |
| PROD(NOP/month) | 4 | 7 | 13 | 25 | 50 |

Calculating productivity of group members,

|  |  |
| --- | --- |
| **Group Members** | **Productivity** |
| Mousumi kumari | 13 |
| Bhagyashri | 13 |
| Ankit Singh Rana | 7 |

Thus PROD= (13+13+7)/3

**PROD = 11**

**Thus, Effort=**141.3/11 **=12.84 PM**

**Calculation of development time**

Calendar time can be estimated using a COCOMO 2 formula

**TDEV = 3 \* (PM) ^ (0.33+0.2\*(B-1.01)),** B is calculated as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Factors** | **Description** | **Rate** | **Rating type** |
| Precedentedness | Reflects the previous experience of the organization with this type of project. Very low means no previous experience, Extra high means that the organization is completely familiar with this application domain. | 4 | Experience of our team is above very low |
| Development flexibility | Reflects the degree of flexibility in the development process. Very low means a prescribed process is used; Extra high means that the client only sets general goals. | 5 | Very low since requirements are clear |
| Architecture/risk resolution | Reflects the extent of risk analysis carried out. Very low means little analysis, Extra high means a complete a thorough risk analysis. | 3 | Nominal |
| Team cohesion | Reflects how well the development team knows each other and work together. Very low means very difficult interactions, Extra high means an integrated and effective team with no communication problems. | 2 | Team cohesion is high |
| Process maturity | Reflects the process maturity of the organization. The computation of this value depends on the CMM Maturity Questionnaire but an estimate can be achieved by subtracting the CMM process maturity level from 5. | 3 | Process maturity is nominal |

PM is the effort computation and B is the exponent computed as discussed above. B is calculated as: **B**= (sum of all factors)/100+1.01

= (17/100) + 1.01

= **1.18**

Assumption, scale is assumed to vary from very Low to extra high (5 to 0).

From the above formula, TDEV= 3\*(12.84) ^ (0.33+0.2\*(1.18-1.01)

= 3\* (12.84) ^ 0.364≈ **7.5 months**

**Staffing requirements**

Staff required can’t be computed by diving the development time by the required schedule. The number of people working on a project varies depending on the phase of the project. The more people who work on the project, the more total effort are usually required.

* In the communication phase, all three team members are needed since what is to be done in the project should be known to all the members.
* In the planning phase, two members are sufficient to decide how the steps are to be carried out and what all works are to be assigned to whom.
* In the modelling phase, minimum one member is needed and maximum two to discuss what are the modules to be included and the language or database to be used.
* In the construction phase, two members are needed one for coding and other for testing the system and removing errors.
* In the deployment phase, one member is sufficient to install the software for the organization for which it is made and train the end users how to use it.

**Actual cost**

# The actual cost of the product is evaluated as =Effort \* cost of one PM

It is assumed that cost of one PM is **15,500 Rs/p-m**

**Actual cost = 12.84\*15,500= Rs 1,99,020**

Therefore, **total cost** = Actual cost+ other costs

|  |  |
| --- | --- |
| Hardware cost | 500 Rs |
| Software cost | 3000 Rs |
| Other (Printer, paper etc) | 2500 Rs |
| Total cost = 6,000 Rs | |

Therefore, total cost = 1,99,020+ 6,000 = **Rs 2,05,020**

# 11.0. RISK MANAGEMENT

## 11.1. Introduction

It involves anticipating risks that might affect the project schedule or the quality of the software being developed and taking action to avoid these risks to ensure that these do not lead to unacceptable budget or schedule slippage. It consists of risk assessment and control.

## 11.2. Basic approach of risk management

All team members assisted in identifying risks in terms of probability and resulting loss. Risk exposure is calculated to prioritize risks to handle them accordingly. Solutions will be provided to manage them. Individual plans will be developed to reduce the exposure of higher priority risks. Specific actions are to be taken within specific time frames. Progress is monitored. As actions are performed, the risk exposure changes, so the priorities continually change.

## 11.3. Risk Assessment

Risk assessment involves measuring the probability that a risk will become a reality. It is a business level decision support tool and a way of gathering the requisite data to make a good judgment call based on knowledge about vulnerabilities**,** threats, impacts, and probability.

## 11.2.1. Risk Identification

Risk identification includes listing project-specific list that are likely to compromise a project’s success. It is the first stage of risk assessment. Risk Identification consists of searching and locating risks before they materialize.

|  |  |  |  |
| --- | --- | --- | --- |
| SL.No. | Risk | Risk Type | Description |
| 1. | Loss of data storage devices | Technical | It might be possible that the data stored might get lost due to damage of hard disk. |
| 2. | The risk associated with computer stoppages or any malfunctioning, defects or software corruption. | Technical | It might be possible that computer stops working may due to sudden crash of hard disk or memory. |
| 3. | The database cannot process as many transactions per second as expected. | Technical | It might be possible that time to retrieve data from database is larger because of large number of sql statements running at a time. |
| 4. | Software is not delivered on time. | Schedule/  Estimation | It may arise due to wrong time estimation,  Resources are not tracked properly. Failure to identify complex functionalities and time required to develop those functionalities. Unexpected project scope expansions, high workload, members suffer from illness. |
| 5. | Size underestimate | Project and product | The size of the software has been underestimated. |
| 6. | Requirements change | Requirement and product | There are changes continuous changes in requirements from time to time. |
| 7. | Inexperienced members | Operational/  People | The group members may be inexperienced, and fail to understand the requirements and have less technical knowledge. |
| 8. | Failure of functionality and performance and quality. | Technical/  Tools | It may happen if product is complex to implement or difficult project modules integration. |
| 9. | Improper design and implementation. | Technical | It may be due to the communication gap between group members, hardware and software requirements are not met within time, less people are available to develop the product. |
| 10. | The degree of uncertainty that the project budget will be maintained. Underestimating the cost drivers. | Estimation | It may arise when there is application complexity, training cost and software corruption. |
| 11. | Hardware and software unavailability. | Project | Hardware that is essential for the project will not be delivered on  Schedule if for some reason it is not available. |
| 12. | Specification delays | Project and product | Specifications of essential interfaces are not available on schedule. |

## 11.2.2. Risk Analysis

In this, each identified risk is considered and a judgement about the probability and the seriousness of it is made. Note: **Risk Factor = Impact factor\* Probability**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl. No. | Risk | Probability | | Impact factor/Loss | Risk Exposure |
| 1. | Loss of data storage devices | 3 | | 8 | 24 |
| 2. | The risk associated with computer stoppages or any malfunctioning, defects or software corruption. | 1 | | 7 | 7 |
| 3. | The database cannot process as many transactions per second as expected. | 1 | | 6 | 6 |
| 4. | Software is not delivered on time. | 7 | | 9 | 63 |
| 5. | Size underestimates. | 4 | | 5 | 20 |
| 6. | Requirements change | 5 | | 7 | 35 |
| 7. | Inexperienced members. | 8 | | 6 | 48 |
| 8. | Failure of functionality and performance and quality. | 7 | 9 | | 63 |
| 9. | Improper design and implementation. | 9 | 8 | | 72 |
| 10. | The degree of uncertainty that the project budget will be maintained. Underestimating the cost drivers. | 6 | 4 | | 24 |
| 11. | Hardware and software unavailability | 4 | 7 | | 28 |
| 12. | Specification delays | 5 | 6 | | 30 |

## 11.2.3. Assumptions

* The impact values are measured **on a scale of 1-10** (1-4 (low), 5(nominal), and 6-10 (catastrophic)).
* The Probability of the risk is measured **on a scale of 1-10** (1-4 (low chance), 5(nominal chance), and 6-10 (high chance)).

**11.2.4. Risk Priotisation**

Once the risks have been identified, they need to be assessed which are more significant. It must depend on risk exposure value. In general, catastrophic risks should always be considered, as should all serious risks that have more than a moderate probability of occurrence.

|  |  |  |  |
| --- | --- | --- | --- |
| SL. NO. | RISK | RISK EXPOSURE | RMP |
| 9. | Improper design and implementation. | 72 | Refer to APPENDIX A page no |
| 8. | Failure of functionality and performance and quality. | 63 |
| 4. | Software is not delivered on time. | 63 |
| 7. | Inexperienced staff | 48 |
| 6. | Requirements change | 35 |
| 12. | Specification delays | 30 |
| 11. | Hardware and software unavailability | 28 |
| 10. | The degree of uncertainty that the project budget will be maintained. Underestimating the cost drivers. | 24 |
| 1. | Loss of data storage devices | 24 |
| 5. | Size underestimates. | 20 |
| 2. | The risk associated with computer stoppages or any malfunctioning, defects or software corruption. | 7 |
| 3. | The database cannot process as many transactions per second as expected. | 6 |

**11.3. Risk Control**

Risk control is the entire process of policies, procedures and systems an institution needs to manage prudently all the risks resulting from its financial transactions, and to ensure that they are within the bank's risk appetite. (Refer to APPENDIX A)

# 12.0. REQUIREMENT ENGINEERING

**12.1. Introduction to requirement engineering process**

The goal of the requirements engineering process is to create and maintain a SRS. It provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification, and managing the requirements as they are transformed into an operational system.

**12.2. Relationships between requirement engineering activities**

The following diagram depicts the relation between requirement engineering activities:

**12.3.** **Requirement elicitation**

In this activity, software engineers work with customers and system end-users to find out about the application domain, what services the system should provide, the required performance of the system, hardware constraints and so on.

**12.3.1. Different elicitation techniques**

**1. Questionnaire:** A questionnaire is a list of written questions that can be completed in one of the two basic ways. First, is a postal questionnaire in which a respondent completes the answers without the help of the researcher. Secondly, respondents could be asked to complete the questionnaire by verbally responding to questions in the presence of the researcher.

Questionnaires are restricted to two basic types of question:

Closed-ended is a question for which a researcher provides a suitable list of responses (e.g. Yes / No). This produces mainly quantitative data. In open-ended, the respondent is asked to answer "in their own words". This produces mainly qualitative data. It is less time-consuming and inexpensive.

**2. Interviews:** It is a (face to face conversation) planned meeting during which you can obtain information from another person. Skills are required to plan, conduct and document interviews. Interviews yield rich insights into people’s biographies, experiences, opinions, values, aspirations, attitudes and feelings. It may be expensive and time consuming.

**3. Focus groups:** A group discussion among groups of people**.** A moderator led in depth on a particular topic or concepts. Get a variety of perspectives/reactions to a certain issue in a short time. It’s a loosely structured, spontaneous discussion managed by a moderator. Focus groups are used for generating information on collective views, and the meanings that lie behind those views.

**4. Observation:** (Naturally observing things in a working environment)The systematic process of recording patterns of occurrences or behaviors - normally without communicating with the people involved.Ask questions about present system operation, observe all steps in the processing cycle, examine each form, record and report, and consider each person working with the system, talk to people who receive current reports.

**5. Document review:** Use document review, to gather background information, review existing system, software documentation, blank copies of forms, samples of completed forms and copies of actual forms.

**6. Research and sampling:** Sampling is a process of collecting examples of actual documents. Its objective is to ensure representation of the overall population accurately, should be considered for interviewing or questionnaires

Research: To study journals, periodicals, books, internet sites, hardware and software vendors, independent firms that provide information, newsgroups, professional meetings, seminars, discussion site visits to observe a system in use.

**12.3.2. Technique used in LMS**

Among the above mentioned data gathering techniques, self-administered questionnaire is usedas a technique for gathering information because at a time, views of a number of users can be collected and it is less time consuming as well as less expensive and more accurate method. Any doubts could be clarified at the moment since it is self-administered. Moreover, it is easily to analyze, less biased and more objective.

**12.3.3. How it is used in LMS?**

For gathering information regarding LMS, following set of questions were distributed among the students and librarian to find out what problems.

**12.3.4. Sample questionnaire**

Dear Respondent, you need to choose from the choices given as they are multiple choice questions. Your individual identity will be treated in **“strict confidential”** and will only be used for research purposes.

**1)** On an average, how many users are registered for issue/return of book?

* 500
* 200
* More than 500
* Other

**2)** Would it be possible that the students too can access their account using the system?

* Yes
* No

**3)** Do want software to be designed or a website for the LMS?

* Website
* Software

**4)** When anyone asks you about how many books are issued and to whom etc, how much effort do you have to put in answering those questions?

* Low
* High
* Very high
* Cannot be predicted

**5)** What all functionalities do you want to be added in LMS?

* Add
* Delete
* Edit
* Any other

**6)** Which of the following will you prefer the most?

* Menus
* Buttons
* Shortcut keys

**7)** Should there be separate report generated for students and books?

* Yes
* No
* Not required

**8)** What should be the color of the interfaces?

* Red
* Purple
* Yellow
* Green

**9)** What will you prefer?

* Metaphors
* Simple buttons
* Both

**12.4. Requirement analysis**

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish.

**12.4.1. User requirements**

It is the high-level abstract requirements written in natural language plus diagrams, of what services are expected to provide and the constraints under which it must operate. It should describe the functional and non-functional requirements so that they are understandable by system users without detailed technical knowledge. They should only specify the external behavior of the system.

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. no | Requirement | Requirem-ent type | Description |
| 1. | Login section | Functional | User shall login the system after he /she enters username and password. |
| 2. | Manage students | Functional | Students shall have their username and password to login into the system and shall be able to view the book details available in the library. |
| 3. | Manage Librarian | Functional | Librarian shall have the highest priority to view details of book/students, add new member or book, search a book/member, edit their information, store and delete data. |
| 4. | Issue book | Functional | Members shall issue book and details shall be shown in their respective account. |
| 5. | Return book | Functional | Members shall return book as per due date and information will be updated accordingly. |
| 6. | Calculate fine | Functional | Fine shall be charged if a member keeps a book longer than the due date. |
| 7. | Generate reports | Functional | Reports should show all necessary information about no of books issued to whom etc. |
| 8. | Constraints on book issue. | Functional | A member shall not issue more than 3 books at a time. |
| 9. | Authentication | Functional | Users need username and password to login into the system. |
| 10. | Student shall only view book details | Non-functional | Student has minimum access rights and shall view only book details. |
| 11. | A book shall be issued for 10 days else fine will be charged. | Non-functional | When a user keeps a book for more than 10 days, fine shall be charged @ 10 Rs/day |
| 12. | Safe and secure | Product | The database may get crashed at any certain time due to virus or OS failure. Therefore, it is required to take the database backup. |
| 13. | Reliable | Product | The software shall be able to accomplish all its tasks in a given time, no failure should occur. |
| 15. | Portable | Product | The system when installed in other systems with fulfilled hardware and software requirements must function in the similar way for which it is developed. |
| 16. | Shall have requisite size, speed, robustness and usability | Product | The software has required number of K bytes space, performs processes in minimum time(less response time), and should have help features and no data corruption or failure of events during run time. |
| 17. | Perform well | Product | This software is not breakdown suddenly in any disaster like power failure. The performance of the functions and every module must be well. The risk factor must be taken at initial step for better performance of the software. |
| 18. | Interoperable | External | The systems shall be able to interact easily with systems on other organizations. |
| 19. | Follow legislation and ethics | External | The system shall operate within law and should be acceptable to its users. |
| 20. | Time constraints | Organizational | The system shall be delivered on 27th October 2012. |
| 21. | Shall follow waterfall methodology | Organizational  Domain | The method followed shall be waterfall and all its stages must be followed. |

**12.4.2. System requirements**

It is the **detailed description** of what the system’s functions, services and operational constraints in detail. It may be part of the contract between the system buyer and the software developers. It should be complete and consistent specification of the whole system.

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No | Requirement | Requirement type | Description |
| 1. | Database: SQL Server 2008 | Software | The back end database shall be sql server 2008. |
| 2. | Operating system: Windows 7 professional-32 or any other version. | Software | The software should be developed in this version of windows and is expected to work in all versions of windows. |
| 3. | Programming language:VB.NET | Software | The language used shall be vb.net as frond end coding. |
| 4. | IDE: Visual studio 2010 | Software | The platform used shall be Visual studio 2010. |
| 5. | Web server: IIS (Internet Information Services (IIS) | Software | IIS is a powerful web server that provides a highly reliable, manageable, and scalable  Web application infrastructure. |
| 6. | RAM size: 3 GB or larger | Hardware | The RAM size of the PC should be 3 GB. |
| 7. | Processor: Intel® core (TM) i3 CPU M350 | Hardware | The processor to be used should be i3. |
| 8. | Hard disk: 20 GB or more | Hardware | The size of the HD should be 20 GB or more. |
| 9. | Display: 1024x768 or higher resolution monitor | Hardware | The monitor should be of minimum this dimension. |

**12.5. Requirements validation**

It is concerned with showing that the requirements actually define the system that the customer wants. It is important because errors in a requirements document can lead to extensive rework costs when they are discovered during development or after the system is in service.

**12.5.1. Validation techniques**

A number of requirements validation techniques can be used in conjunction or individually:

**1. Requirements reviews:** The requirements are analyzed systematically by a team of reviewers.

**2. Prototyping:** An executable model of the system is demonstrated to end-users and customers who can see if it meets their real needs. A prototype is an initial version of a software system that is used to demonstrate concepts, try out design solutions and find out more about the problem and its possible solutions. It helps in controlling costs.

**3. Test-case generation:** Requirements should be testable to reveal requirements problems. If a test is difficult to design, it means that it would be difficult to implement and should be reconsidered.

**12.5.2. Validation technique used for LMS**

The validation technique used for LMS is prototyping.

**12.5.2.1. List of reviewers chosen for LMS**

Reviewers chosen for validating requirements are the stakeholders like students and librarian.

|  |  |  |
| --- | --- | --- |
| Sl. No. | Requirement | Comment |
| 1. | Login section | The screen design is good, but you can give the users to recover password. |
| 2. | Manage students | Student id should be autogenerated. Search should be done more efficiently. Student could change their password themselves. |
| 3. | Manage Librarian | It is working well. But multiple deletion , updation could have been done at a time. |
| 4. | Issue book | Issue book screen is not so interactive, it could have been made more functional. |
| 5. | Return book | Students should get prompted when to return books. Provision for returning more than 1 book at a time. |
| 6. | Calculate fine | Record could be kept if a student wants only to know his fine. |
| 7. | Generate reports | Reports generation is not sufficient. |
| 8. | Constraints on book issue. | It is working well. |
| 9. | Authentication | Strong authentication should be done. |
| 10. | Student shall only view book details | The form design is not up to the mark. Student has difficulty locating things. Help feature should be included. |
| 11. | A book shall be issued for 10 days else fine will be charged. | Yes, it is highly efficient. |
| 12. | Safe, secure, reliable and portable. | Sometimes it is unreliable. |
| 16. | Shall have requisite size, speed, robustness and usability | The interface it not much user-friendly. |
| 18. | Interoperable | - |
| 19. | Follow legislation and ethics | The system follows legislation and ethics. |

# 13.0 Process Modelling

## 13.1. Introduction

A formal way of representing how data flows, processes execute and data stored or retrieved within the given system with the help of set of coherent, interrelated data flow diagrams. It comprises two types of modeling: Logical Data Modeling and Physical Data Modeling

In the development of Library Management System, only logical process modeling is taken into consideration.

## 13.2. Logical Data Modelling

## A logical data model is a way of graphically representing what the information is really all about, how it relates to other information and business concepts, and how business rules are applied to its use in the system. The logical data modeling comprises following representations of the processes that are carried out stepwise in the LMS:

## 13.2.1. Context Diagram

A context diagram is a top level data flow diagram that shows the boundaries or scope of a particular system. It contains only one process node (process 0) that generalizes the function of the entire system in relationship to external entities. In the LMS, the developer has assumed four external entities viz. , student, librarian, book, publisher as shown below:

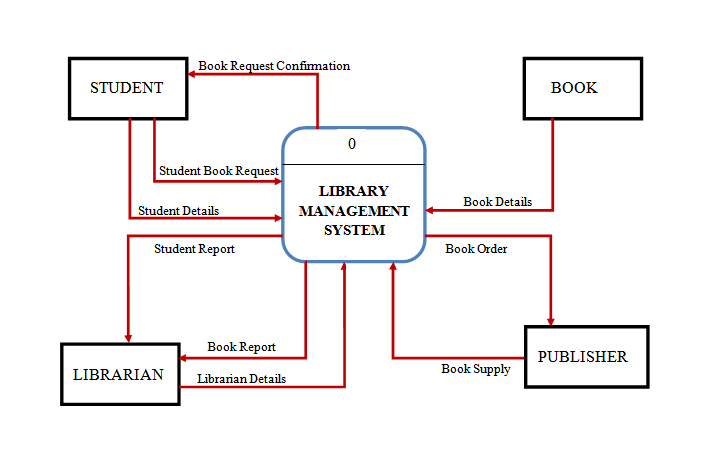


Figure Context Diagram

**13.2.2 Functional Decomposition**

An iterative hierarchical process of refining the details of a system, creating a set of charts at lower and lower levels, in which a process at a certain level is explained on the next level in greater detail. It is done till we get Primitive DFD i.e., the lowest level DFD, where no process can be broken any further. Fn Decomposition for LMS is given below:

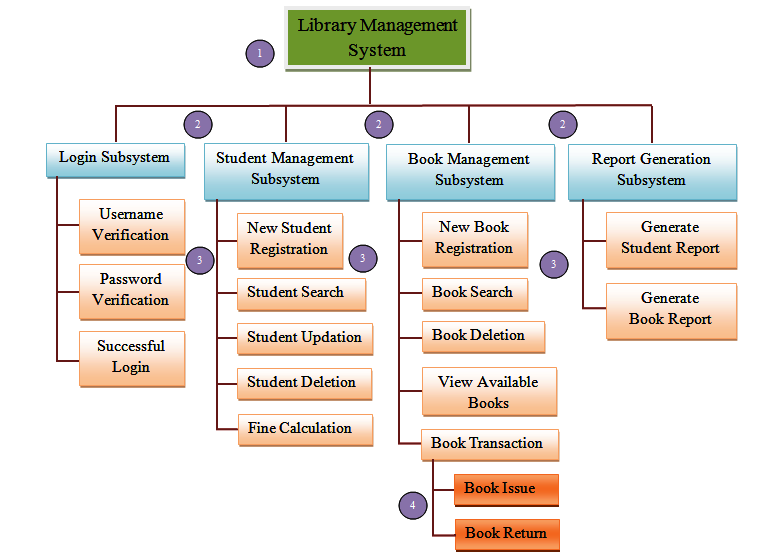


Figure 2 Functional Decomposition of LMS

**13.2.3 Level 0 DFD**

Level 0 is a data flow diagram that zooms in on the context diagram to give a more detailed view as shown below. It shows four major processes viz., user login, manage student, manage book and generate report in case of LMS.

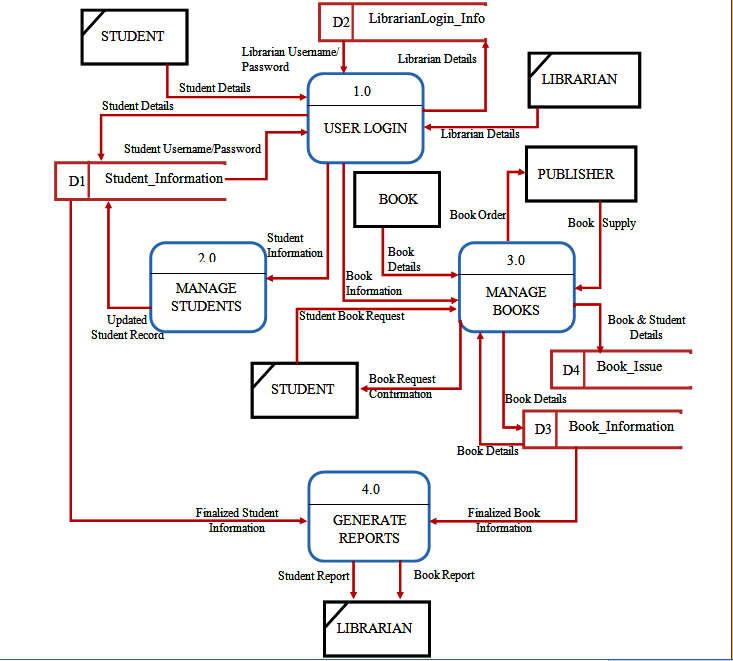


Figure Level 0 DFD for LMS

**13.2.4. Level 1 DFD**

A specific process from level 0 DFD is further split into number of sub processes that performs their unique function. The developer has considered four main level 1 processes for the LMS as shown in the following figure.

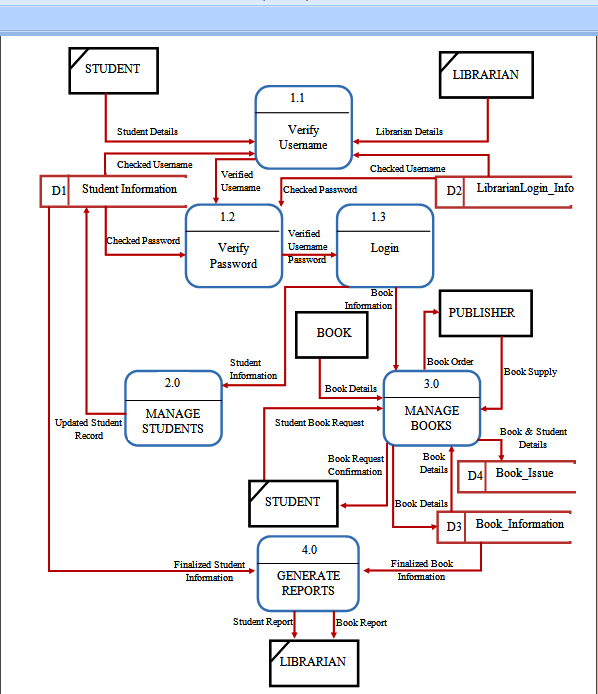


Figure Level 1 of process 1.0 User Login

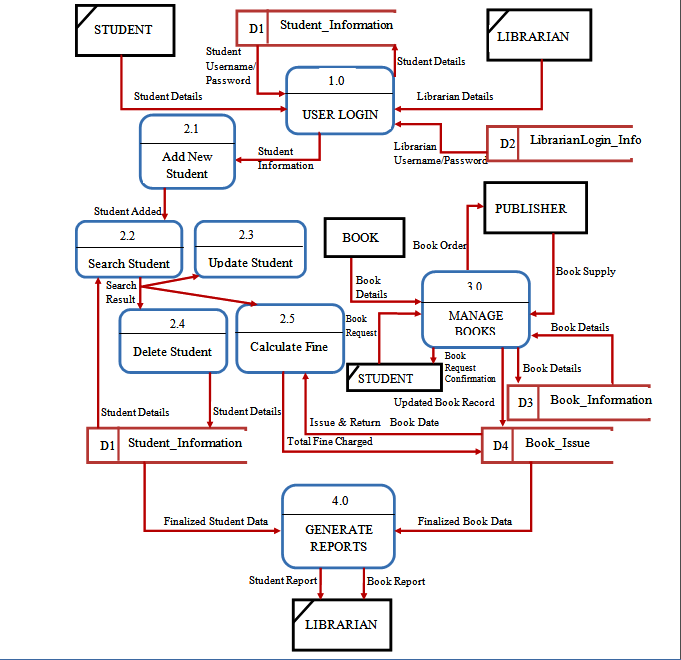


Figure Level 1 of process 2.0 Manage Students

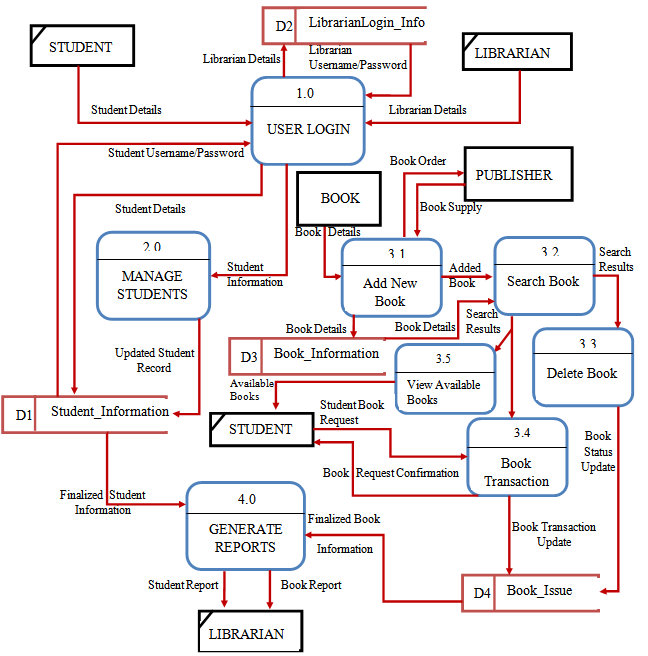


Figure Level 1 DFD of process 3.0 Manage Books

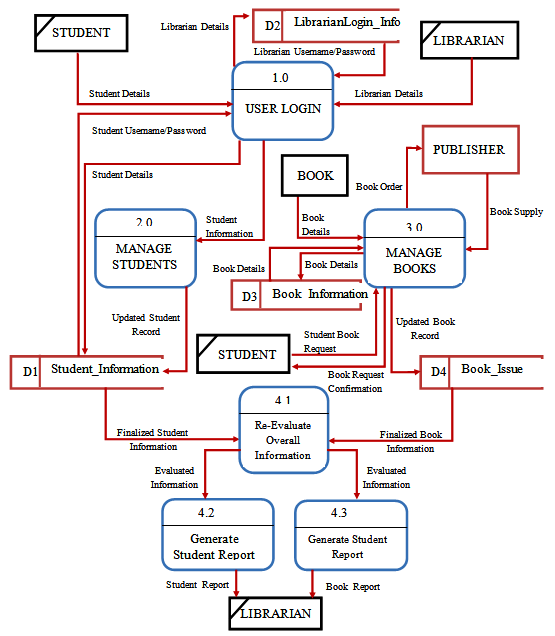


Figure Level 1 DFD of process 4.0 Generate Reports

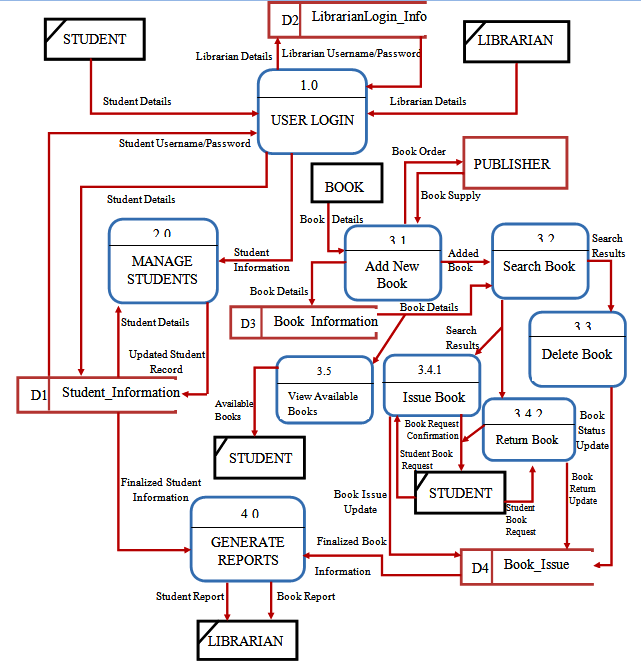


Figure Level 2 DFD of process 3.4 Book Transaction

**13.2.5. Process Specification**

**13.2.2.5.1 Structured English**

Structured English is a subset of standard English that describes logical processes clearly and accurately. For LMS, structured English is represented in the following way:

|  |  |
| --- | --- |
| **Process 1.0: USER LOGIN**  DO  READ user type  IF user type selected is student THEN  View details of book available  ELSEIF user type selected is librarian THEN  Manage books or manage students or Generate reports  UNTIL End-of-File | **Process 4.0: GENERATE REPORTS**  DO  READ report selected  IF student report is selected THEN  Display student report  ELSEIF book report is selected THEN  DISPLAY book report  UNTIL End-of-File |
| **Process 2.0: MANAGE STUDENTS**  DO  READ option selected  IF add new student is selected THEN  READ student details  STORE in student\_information  ELSEIF search student is selected THEN  SEARCH student by its intake number  DISPLAY its information  ELSEIF update student is selected THEN  SEARCH the student by its intake number  UPDATE details  STORE details in student\_information  ELSEIF delete student is selected THEN SEARCH the student by its intake number  UPDATE its status as deactivated  ELSEIF calculate fine is selected THEN  SEARCH the student by its intake number  READ no of books issued, dates when issued, dates when returned  From book\_issue table  CALCULATE fine= No. of Days \* 10 Rs. (i.e. 10 Rs fine per day)  DISPLAY calculate fine  UNTIL End-of-File | **Process 3.0: MANAGE BOOKS**  DO  READ option selected  IF add new book is selected THEN  READ book details  STORE in book information  ELSEIF search book is selected THEN  SEARCH book by its book code no  DISPLAY all its information  ELSEIF delete book is selected THEN Search book by its book code no UPDATE its status as deleted in book information  ELSEIF view available book is selected THEN  DISPLAY list of all available books  ELSEIF book transaction is selected THEN  IF book issue is selected THEN  READ book code no, student id no, book issue date  STORE this in book issue  DISPLAY book issued successfully  ELSEIF book return is selected THEN  READ book code no, student id no, book return date  STORE this in book issue  DISPLAY book returned successfully  UNTIL End-of-File |

**13.2.2.5.2 Decision Tree**

A decision tree is a graphical representation of the conditions, actions, and rules. It shows the logic structure in a horizontal form that resembles a tree with the roots at the left and the branches to the right. Decision tree for different processes of LMS is shown below:

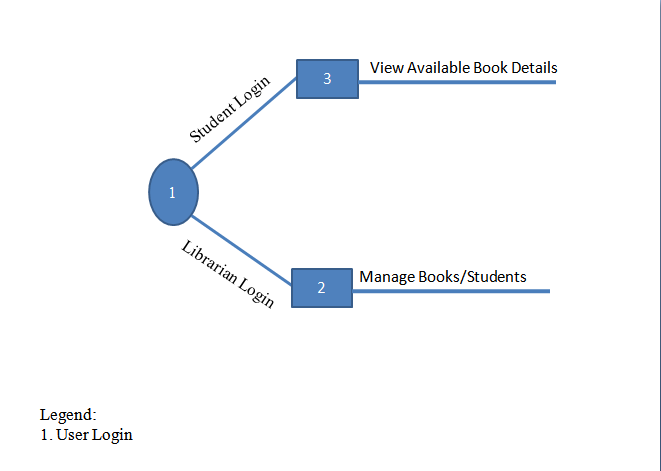


Figure Decision Tree for User Login process

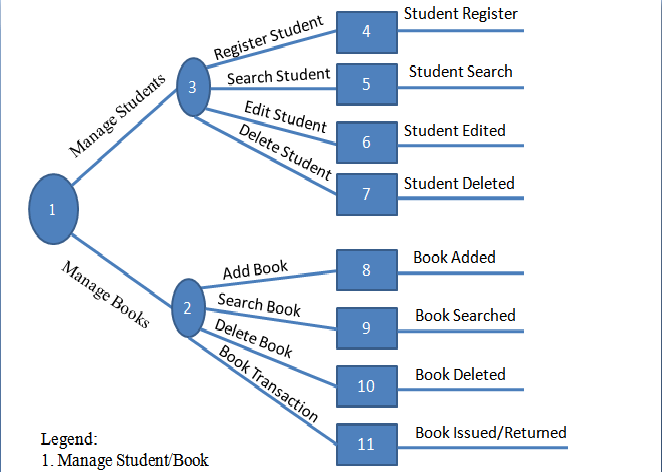


Figure Decision Tree for Manage Student/Book process

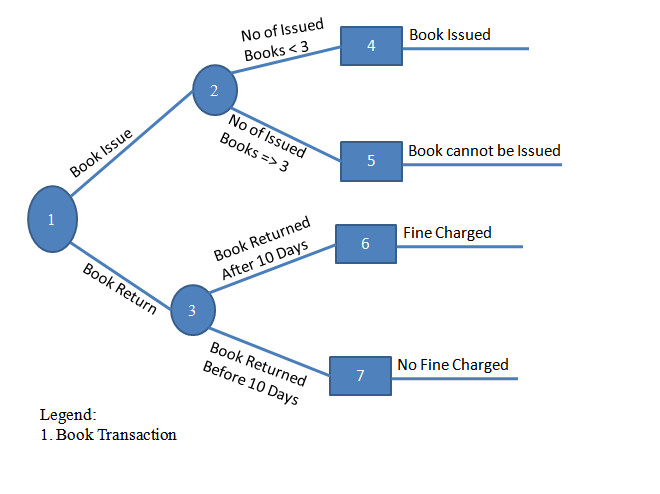


Figure Decision Tree for Book Transaction process

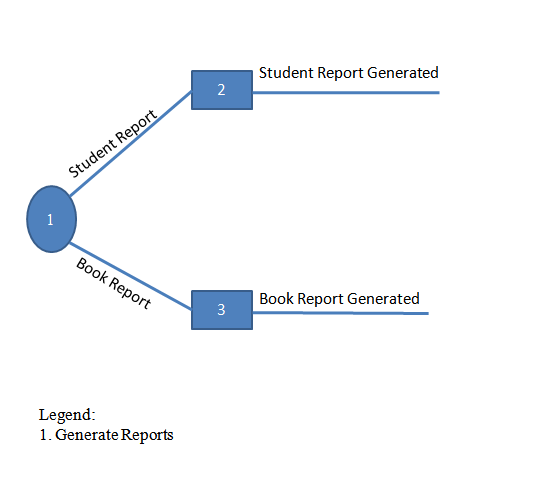


Figure Decision Tree for Generate Report process

# 14.0 Data Modelling

**14.1. Logical Data Models**

**14.1.1.** Entity Relationship Diagram

An entity-relationship (ER) diagram is a specialized graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Rectangular boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. ER diagram for LMS is shown below:

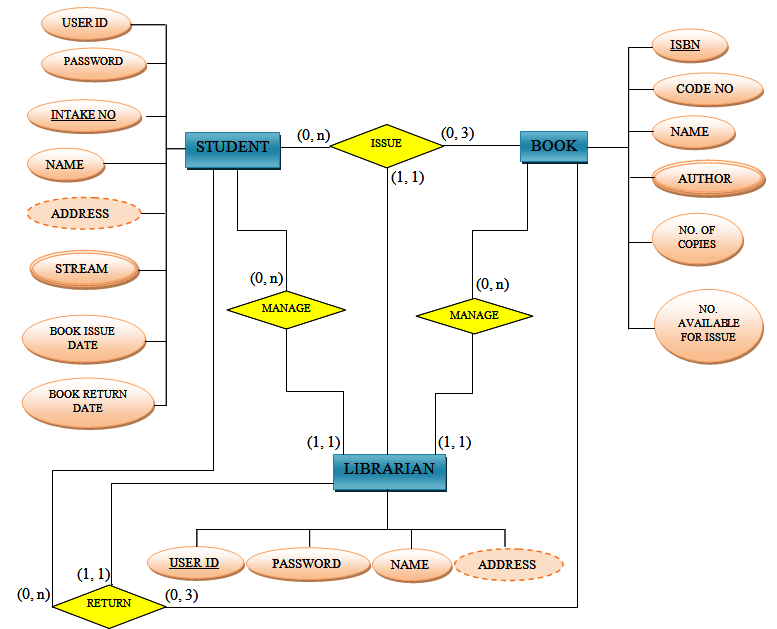
****

Figure Entity Relationship Diagram for LMS

**14.1.2. Entity Life History**

Entity life history (ELH) diagram represents the life cycle of entities within the database. In the database of LMS, three entities for example Student, Book and Librarian are assumed and are illustrated below:

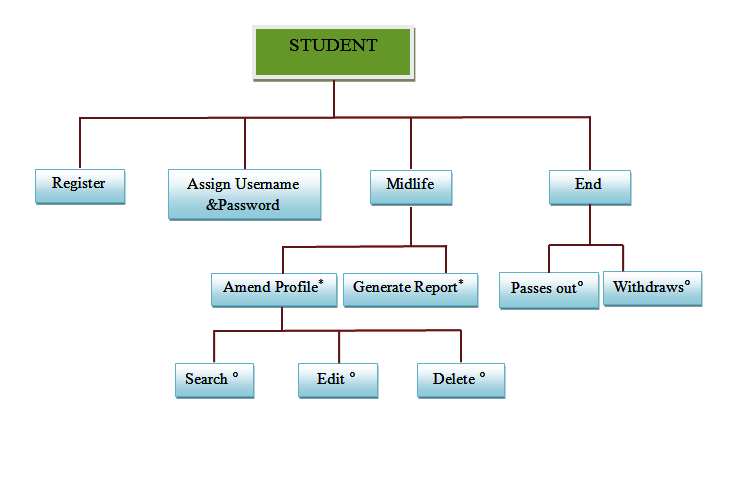


Figure Entity Life History of Student

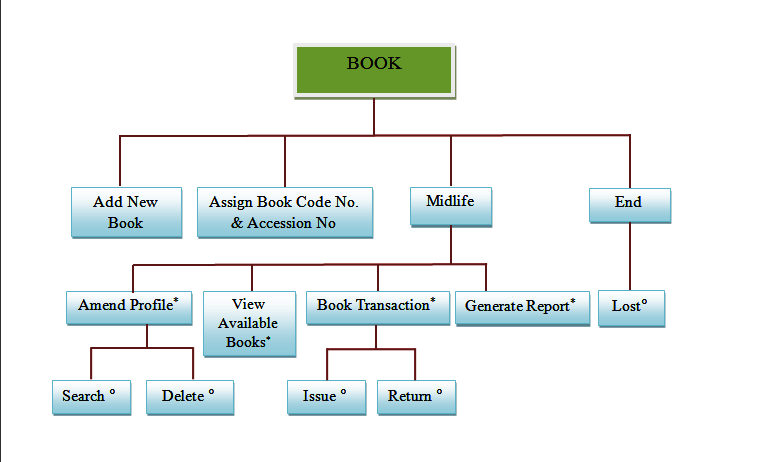


Figure Entity Life History of Book

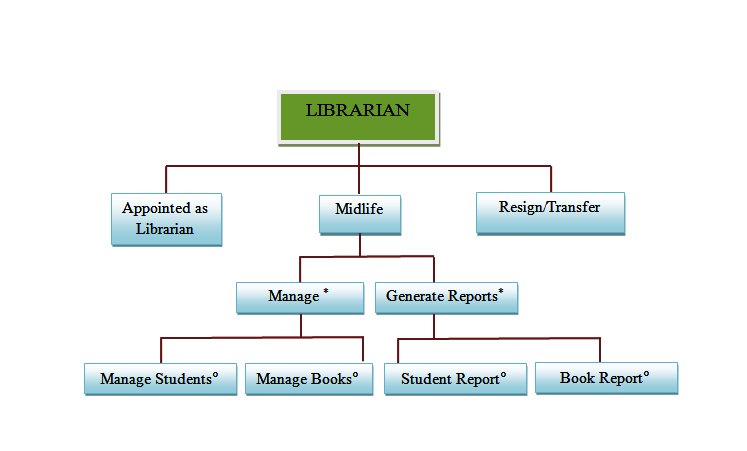


Figure Entity Life History of Librarian

# 15.0 Data Dictionary for Proposed Systems

A data dictionary or data repository is a central storehouse of information about the system’s data. It is used to collect, document, and organize specific facts about the system, including the contents of data flows, data stores, external entities and processes. (Note: Since there is no alternative names for the data flows, processes, data store and entities, Alias filed in the definition is not given.)

**15.1. Definition of Data flows are as follows:**

|  |  |
| --- | --- |
| Name | Student Details |
| Origin | STUDENT external entity |
| Destination | USER LOGIN process |
| Content Description | Student Details = [Student\_Id + Student\_Password]  Student\_Id=prefix + serial no  Prefix=ST  Serial no = \*any five digit number string\*  Student\_Password=\*any number or string\* |
| Supplementary  Information | Student\_Id is given as student username and Student\_Password as its password by the student to log in. |

|  |  |
| --- | --- |
| Name | Student Username/Password |
| Origin | Student\_Information data store |
| Destination | USER LOGIN process |
| Content Description | Student Username/Password = [Student\_Id + Student\_Password]  Student\_Id=prefix + serial no  Prefix=ST  Serial no = \*any five digit number string\*  Student\_Password=\*any number or string\* |
| Supplementary  Information | Student\_Id is given as student username and Student\_Password as its password by the student to log in. |

|  |  |
| --- | --- |
| Name | Librarian Username/Password |
| Origin | LibrarianLogin\_Info data store |
| Destination | USER LOGIN process |
| Content Description | Librarian Username/Password = Librarian \_Id + Librarian\_Password |
| Supplementary Information | Librarian \_Id is given as Librarian username and Libraria\_Password as its password by the Librarian to log in. |

|  |  |
| --- | --- |
| Name | Librarian Details |
| Origin | LIBRARIAN external entity |
| Destination | USER LOGIN process |
| Content Description | Librarian Details = Librarian \_Id + Librarian\_Password |
| Supplementary Information | Librarian \_Id is given as Librarian username and Libraria\_Password as its password by the Librarian to log in. |

|  |  |
| --- | --- |
| Name | Student Information |
| Origin | USER LOGIN process |
| Destination | MANAGE STUDENTS process |
| Content Description | Student Information = Student\_Id +  Email\_id + Student\_Name + Father\_Name + Mother\_Name + Gender + DateofBirth + Address + Phone\_Number + Class + Account\_Status |
| Supplementary Information | Complete information about the student. |

|  |  |
| --- | --- |
| Name | Updated Student Record |
| Origin | MANAGE STUDENTS process |
| Destination | Student\_Information datastore |
| Content Description | Updated Student Record = Student\_Id [|]Email\_id [|]Student\_Name [|]Father\_Name [|]Mother\_Name [|]Gender [|]DateofBirth [|]Address [|]Phone\_Number [|]Class [|]  Account\_Status |
| Supplementary Information | Update any of the information about the student given above. |

|  |  |
| --- | --- |
| Name | Book Order |
| Origin | MANAGE BOOKS process |
| Destination | PUBLISHER external entity |
| Content Description | Book Order=Book\_Name +Book\_Author +Book\_No\_Of\_Copies |
| Supplementary Information | Information about ordering the Book. |

|  |  |
| --- | --- |
| Name | Book Details |
| Origin | BOOK external entity |
| Destination | MANAGE BOOKS process |
| Content Description | Book Details= Book\_Code +Book\_Name +Book\_Category +ISBN\_NN +Book\_Author +Book\_Edition+Book\_Price +Book\_publisher +Book\_No\_Of\_Copies +  No\_of\_Pages +Book\_No\_Available |
| Supplementary Information | Complete information about the Book. |

|  |  |
| --- | --- |
| Name | Book Information |
| Origin | USER LOGIN process |
| Destination | MANAGE BOOKS process |
| Content Description | Book Information = Book\_Code +Book\_Name +Book\_Category +ISBN\_NN +Book\_Author +Book\_Edition+Book\_Price + Book\_publisher +Book\_No\_Of\_Copies + No\_of\_Pages + Book\_No\_Available |
| Supplementary Information | Complete information about the Book. |

|  |  |
| --- | --- |
| Name | Book & Student Details |
| Origin | MANAGE BOOKS process |
| Destination | Book\_Issue data store |
| Content Description | Book & Student Details =  Accession\_No +Book\_Id +Issue\_Date +  Return\_Date +Allocated\_Student\_Id +  Fine +Return\_Status |
| Supplementary Information | Information about the Book as well as student that has been issued to the student. |

|  |  |
| --- | --- |
| Name | Book Supply |
| Origin | PUBLISHER external entity |
| Destination | MANAGE BOOKS process |
| Content Description | Book Supply=Book\_Name +Book\_Author +Book\_No\_Of\_Copies +Book\_Edition+Book\_Price +ISBN\_NN |
| Supplementary Information | Information about Book Supply. |

|  |  |
| --- | --- |
| Name | Book Request Confirmation |
| Origin | MANAGE BOOKS process |
| Destination | Student external entity |
| Content Description | Student Book Request = Book\_Name [|]  Book\_Id+ Issue\_Date +Return\_Date |
| Supplementary Information | Confirmation of either issuing a book or returning a book. |

|  |  |
| --- | --- |
| Name | Finalized Student Information |
| Origin | Student\_Information datastore |
| Destination | GENERATE REPORTS process |
| Content Description | Finalized Student Information = Student\_Id +  Email\_id + Student\_Name + Father\_Name + Mother\_Name + Gender + DateofBirth + Address + Phone\_Number + Class + Account\_Status |
| Supplementary Information | Complete information about the student. |

|  |  |
| --- | --- |
| Name | Student Book Request |
| Origin | Student external entity |
| Destination | MANAGE STUDENTS process |
| Content Description | Student Book Request = Book\_Name [|]  Book\_Id |
| Supplementary Information | Request for either issuing a book or returning a book. |

|  |  |
| --- | --- |
| Name | Finalized Book Information |
| Origin | Book\_Information datastore |
| Destination | GENERATE REPORTS process |
| Content Description | Finalized Book Information = Book\_Code +Book\_Name +Book\_Category +ISBN\_NN +Book\_Author |
| Supplementary Information | Complete information about the book required to generate report. |

|  |  |
| --- | --- |
| Name | Student Report |
| Origin | GENERATE REPORTS process |
| Destination | Librarian external entity |
| Content Description | Student Report= Student\_Id + Student\_Name + + Class + Account\_Status |
| Supplementary Information | A final student report is generated with the necessary details. |

|  |  |
| --- | --- |
| Name | Book Report |
| Origin | GENERATE REPORTS process |
| Destination | Librarian external entity |
| Content Description | Book Report= Student\_Id + Student\_Name + Class +Account\_Status |
| Supplementary Information | A final book report is generated with the necessary details. |

**15.2. Definitions of Data Stores are as follows:**

|  |  |
| --- | --- |
| Name | LibrarianLogin\_Info |
| Input Data Flows | Librarian Details |
| Output Data Flows | Librarian Username/Password |
| Content Description | LibrarianLogin\_Info = Librarian\_Id +  Librarian\_Password +  Description +  Account\_Status |
| Supplementary Information | Username is stored as Librarian\_Id and password as Librarian\_Password. |

|  |  |
| --- | --- |
| Name | Book\_Information |
| Input Data Flows | Book Details |
| Output Data Flows | Book Details, Finalized Book Information |
| Content Description | Book\_Information = Book\_Code +  Book\_Name +Book\_Category +ISBN\_NN +Book\_Author +Book\_Edition+Book\_Price +Book\_publisher +Book\_No\_Of\_Copies +  No\_of\_Pages +Book\_No\_Available |
| Supplementary Information | Complete information about the Book is stored in Book\_Information data store. |

|  |  |
| --- | --- |
| Name | Student\_Information |
| Input Data Flows | Student Details, Updated Student Record |
| Output Data Flows | Student Username/Password, Finalized Student Information |
| Content Description | Student\_Information = Student\_Id +  Email\_id +Student\_Nam+Father\_Name +Mother\_Name +Gender +DateofBirth +Address +Phone\_Number +Class  +Account\_Status |
| Supplementary Information | Complete information about the student stored in Student\_Information table. |

|  |  |
| --- | --- |
| Name | Book\_Issue |
| Where/How | Used to store the book issue information about the book and student at the time of issuing a book to the student. |
| Content Description | Book\_Issue = Accession\_No +  Book\_Id +Issue\_Date +Return\_Date +  Allocated\_Student\_Id +Fine +Return\_Status |
| Supplementary  Information | Contains the book issue/return information. |

**15.3. Definitions of Processes are as follows:**

|  |  |
| --- | --- |
| Name | User Login |
| Input Data Flows | Librarian Username/Password, Student Username/Password, Student Details, Librarian Details |
| Output Data Flows | Book Information, Student Information |
| Process Description  For each Librarian username/password OR Student username/password  Display admin page OR student page  Output Book Information, Student Information | |
| Supplementary Information | Process used to login into LMS |

|  |  |
| --- | --- |
| Name | Manage Books |
| Input Data Flows | Book Information, Book Details, Book Supply, Student Book Request |
| Output Data Flows | Book & Student Details, Book Details, Book Request Confirmation |
| Process Description  For each Book Information  Store in Student\_Information table  Output Book & Student Details, Book Request Confirmation | |
| Supplementary Information | Process used to add, search, delete, issue, return book. |

|  |  |
| --- | --- |
| Name | Manage Students |
| Input Data Flows | Student Information |
| Output Data Flows | Updated Student Record |
| Process Description  For each Student Information  Store in Student\_Information table  Output Updated Student Record | |
| Supplementary Information | Process used to add, search, update, delete, calculate fine of the student. |

|  |  |
| --- | --- |
| Name | Generate Reports |
| Input Data Flows | Finalized Student Information, Finalized Book Information |
| Output Data Flows | Student Report, Book Report |
| Process Description  For each Finalized Student Information or Finalized Book Information  Retrieve Student Information or Book Information  Output Student Report or Book Report | |
| Supplementary Information | Process used to generate Reports of Students and Books. |

1. **15.4. Definitions of Source and Sink** **are as follows:**

|  |  |
| --- | --- |
| Name | STUDENT |
| Input Data Flows | Book Request Confirmation |
| Output Data Flows | Student Details, Student Book Details |
| Content Description | Student = [Student\_Id + Student\_Password] + Email\_id +  Student\_Name + Father\_Name + Mother\_Name + Gender + DateofBirth + Address + Phone\_Number + Class +  Account\_Status  Student\_Id=prefix + serial no  Prefix=ST  Serial no = \*any five digit number string\*  Student\_Password=\*any number or string\* |
| Supplementary Information | Attributes relevant to student are mentioned. |

|  |  |
| --- | --- |
| Name | BOOK |
| Input Data Flows | - |
| Output Data Flows | Book Details |
| Content Description | Book = Book\_Code +Book\_Name +Book\_Category +ISBN\_NN +Book\_Author +Book\_Edition+ Book\_Price + Book\_publisher +Book\_No\_Of\_Copies + No\_of\_Pages + Book\_No\_Available |
| Supplementary Information | Attributes relevant to Book are mentioned. |

|  |  |
| --- | --- |
| Name | LIBRARIAN |
| Input Data Flows | Student Report, Book Report |
| Output Data Flows | Librarian Details |
| Content Description | Librarian = Librarian\_Id +  Librarian\_Password +Description +  Account\_Status |
| Supplementary Information | Attributes relevant to Librarian are mentioned. |

|  |  |
| --- | --- |
| Name | PUBLISHER |
| Input Data Flows | Book Order |
| Output Data Flows | Book Supply |
| Content Description | Publisher= Book\_publisher |
| Supplementary Information | Attributes relevant to Publisher are mentioned. |

# 16.0. DESIGN PRINCIPLES AND CONCEPTS

**16.1. Introduction to design**

It is the first step in the development phase .Final blueprint of software is a software design. It is a representation of a product or a system with sufficient detail for implementation. It consists of mapping the information from the analysis model to the design representations. Input of software design is requirement analysis models and specification document and output of software design is design models and design specification document. Design translates the requirements into a completed design model for a software product, provides the representations of software that can be assessed for quality.

**16.1.1. The design model**

The design model consists of:

* **Data Design:** Transforms information domain model into data structures required to implement software. The data objects and relationships defined in the ERD and the detailed data content depicted in the data dictionary provide the basis for the data design activity. For LMS, the data design is as follows:

For Book\_Information, Student\_Information

|  |  |  |
| --- | --- | --- |
| Sl. No. | Fields | Type and length |
| 1. | Book\_Code | varchar(50) |
| 2. | Book\_Name | varchar(50) |
| 3. | Book\_Category | varchar(50) |
| 4. | ISBN\_NN | varchar(50) |
| 5. | Book\_Author | varchar(50) |
| 6. | Book\_Edition | varchar(10) |
| 7. | Book\_Price | float |
| 8. | Book\_publisher | varchar(50) |
| 9. | Book\_No\_Of\_Copies | int |
| 10. | No\_of\_pages | int |
| 11. | Book\_No\_Available | int |

|  |  |  |
| --- | --- | --- |
| Sl. No. | Fields | Type and length |
| 1. | Student\_ID | varchar(50) |
| 2. | Student\_Password | varchar(50) |
| 3. | Email\_Id | varchar(100) |
| 4. | Student\_Name | varchar(50) |
| 5. | Father\_Name | varchar(50) |
| 6. | Mother\_Name | varchar(50) |
| 7. | Gender | varchar(50) |
| 8. | Date\_of\_Birth | Date |
| 9. | Address | varchar(Max) |
| 10. | Phone\_Number | varchar(15) |
| 11. | Class | varchar(15) |
| 12. | Account\_Status | varchar(50) |

For Book\_Issue and LibrarianLogin\_Info tables

|  |  |  |
| --- | --- | --- |
| Sl. No. | Fields | Type and length |
| 1. | Accession\_No | varchar(50) |
| 2. | Book\_Id | varchar(50) |
| 3. | Issue\_Date | varchar(50) |
| 4. | Return\_Date | varchar(50) |
| 5. | Allocated\_Student\_Id | varchar(50) |
| 6. | Fine | float |
| 7. | Return\_Status | bit |

|  |  |  |
| --- | --- | --- |
| Sl. No. | Fields | Type and length |
| 1. | Librarian\_Id | varchar(50) |
| 2. | Librarian\_Password | varchar(50) |
| 3. | Description | varchar(Max) |
| 4. | Account\_Status | varchar(50) |

* **Architectural Design:** Refer to ‘Architectural Design’ section.
* **Interface Design:** Refer to ‘Interactive Screen Design’ section.
* **Procedural Design:** Transforms structural elements of the architecture into a procedural description of software construction. For LMS, the procedural design is as follows:

|  |  |  |
| --- | --- | --- |
| SL. No. | Name of the procedures and events | Description |
| 1. | btnLogin\_Click | This procedure is in the Login form. On clicking it, the username and password entered by the user is matched by that in the database. |
| 2. | btnAdd\_Click | This is in the AddBook\_frm form. On clicking it, it is checked whether the user has entered book details in correct format or not. If yes, new book is added else proper message is prompted. |
| 3. | btnSearch\_Click | It is in the Updatebook form. On clicking it, all the books related to searched entry is displayed in the table. |
| 4. | btnUpdate\_Click | It is in the Updatebook form. On clicking it, changes made in the table is updated in the book\_information database. |
| 5. | btnDelete\_Click | It is in the Updatebook form. It allows the user to delete the information of the related book to be permanently deleted from the database. |
| 6. | bnAdd\_Click | It is in the Registermember form. On clicking it, it is checked whether the user has entered student details in correct format or not. If yes, new student is registered else proper message is prompted. |
| 7. | btnSearch\_Click | It is in the UpdateUser form. On clicking it, information of students related to the searched entry is displayed in the table. |
| 8. | btnUpdate\_Click | It is in the UpdateUser form. On clicking it, changes made in the table is updated in the student\_information database. |
| 9. | btnDelete\_Click | It is in the UpdateUser form. It allows the user to delete the information of the selected student to be permanently deleted from the database. |
| 10. | btnEnter\_Click | It is in the Bookissue form. On pressing it, the details of issued book is displayed in the table from book\_information database. |
| 11. | btnIssue\_Click | It is in the Bookissue form.Book is issues to the student (information goes to boo\_issue table) if he/she has less than 3 books, else a message is prompted to issue another book. |
| 12. | btnReturn\_Click | It is in the Bookreturn form. Book is returned and return\_ststus becomes true here, fine is also calculated and displayed. |
| 13. | btnEnter\_Click | It is in the Bookreturn form. It displays list of all books which are returned or are to be returned. |
| 14. | btnSearch\_Click | It is in the SearchBookByStudent form. On clicking it, all the available books in the library is displayed in the table. |

**16.2. Software design** **principles**

* **Design process should not suffer from “tunnel vision”**- i.e. there should be no restriction on requirements. In LMS, all the requirements given by the end users are completely satisfied and not confined. For e.g., the design team considered alternative approaches based on the requirements of the problem and resources available.
* **The design should be traceable to the analysis model**- since, single element of design model traces to multiple requirements. In LMS all that are followed in analysis modelling is mapped to the design model with no addition or removal of things. The data store used in DFD is the same table for which data design has been made.
* **The design should not reinvent the wheel; Time is short** – New things should not be introduced in the design part. In LMS, while designing, the design patterns were chosen from already design patterns so that the final product is not something reinvented than the required one.
* **The design should “minimize intellectual distance” between the software and the problem in the real world-** It means that, the difference between (real-world problem and the computerized solution to the problem) should not be much. LMS is so designed so that it is fully able to solve the real world problem.
* **The design should exhibit uniformity and integration:** A design is uniform if it appears that one person developed the whole thing. Thus, rules of style and format were defined by the design team before design work began. A design is integrated if care is taken in defining interfaces between design components.
* **The design should be structured to accommodate change:** The LMS developed is very flexible to accommodate changes in future**.**
* **The design should be structured to degrade gently:** even when aberrant data, events, or operating conditions are encountered.Well-designed software should never “bomb”. The design team has added all features to accommodate unusual circumstances, and if necessary it must terminate processing in a graceful manner.
* **Design is not coding, coding is not design:** Design is as important as coding as design decisions made of the coding level address the small implementation details that enable the procedural design to be coded. For LMS, designing principles and concepts are all done before starting the programming environment as specified in the documentation.
* **The design should be reviewed to minimize conceptual (semantic) errors.** The design team reviewed many time for minimizing the conceptual errors (inconsistency, ambiguity etc).

**16.3. Software design** **concepts**

* **Abstraction** – Hiding unnecessary detail intentionally and showing only the necessary detail or concentrate on a problem at some level of generalization without regard to irrelevant low level details. It is of three types, data, procedural and control abstraction. For e.g. only interfaces will be shown to the end users with no coding detail (how it is done) in LMS.
* **Refinement** - Process of elaboration where we have provided successively more lower-level detail for each design component. Abstraction and refinement are complementary concepts. It is a top-down design strategy. The design team has given the detail for each component of the system from highest level to lowest possible level. E.g. in report generation, lower level detail is shown.
* **Modularity** – (Divide and conquer) the degree to which a software can be understood by examining its components independently of one another. Modularity helps in system debugging isolating the system problem to a component is easier. The LMS is divided into 5 modules as discussed earlier.
* **Software architecture** – Software architecture is the hierarchical structure of program components (modules), the manner in which these components interact and the structure of data that are used by the components.. Overall structure of our software components and the ways in which that structure provides conceptual integrity for our system. For LMS, architectural design is discussed later in diagram.
* **Control Hierarchy**- Control hierarchy, also called program structure, represents the organization of program components (modules) and implies a hierarchy of control. It does not represent procedural aspects of software. In LMS, the librarian module controls the book and student module.
* **Structural partitioning** - Horizontal partitioning defines three partitions (input, data transformations, and output); In LMS, control decisions in top level modules and processing work in the lower level modules.
* **Data structure** – the logical relationship among individual data elements. The system has been designed in such a way that we have mentioned the relationship between the data elements.
* **Software procedure** - Processing details of individual modules. In LMS, precise specification of processing (event sequences, decision points, repetitive operations, data organization/structure) is done for each module.
* **Information hiding** - It hides those program components from other classes which are likely to change. E.g. Information (data and procedure) contained in one module is inaccessible to modules that have no need for such information. Thus in LMS, information hiding is achieved by sending only necessary information between 5 the modules.

**Design models**

**Data design:** It represents data objects used by the software. Transforms information domain model into data structures required to implement software.

# 17.0. ARCHITECTURAL DESIGN

**17.1. Explanation**

Architectural design represents the structure of data and program components that are required to build a computer-based system. It considers the architectural style that the system will take, the structure and properties of the components that constitute the system, and the interrelationships that occur among all architectural components of the system.

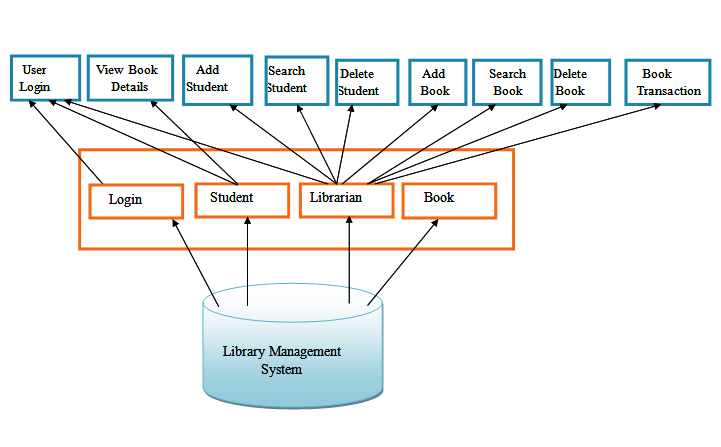
****

Figure 17 Architectural Design for LMS

**17.2. Justification:**

Representations of software architecture are an enabler for communication between all parties interested in the development of a computerized Library Management System. The architectural design highlights early design decisions that will have a profound impact on a all software engineering work that follows and, as important, on the ultimate success of the system as an operational entity. Architecture “constitutes a relatively small, intellectually graspable model of how the system is structured an how its components work together ”[Bas03].

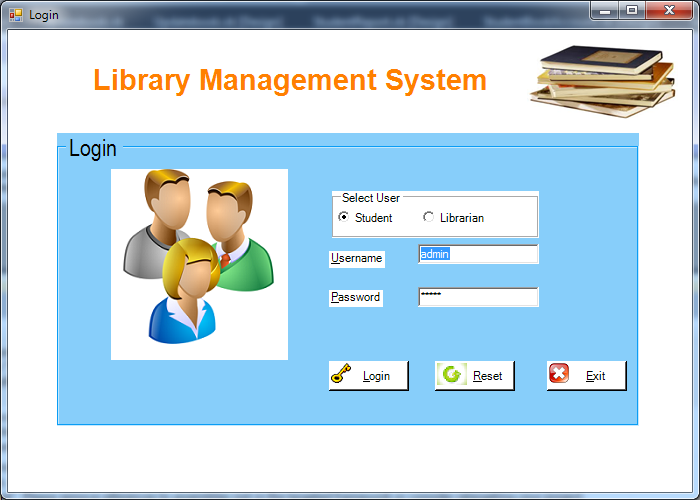
# 18.0. INTERACTIVE SCREEN DESIGNS

**18.1. Input and output screens**

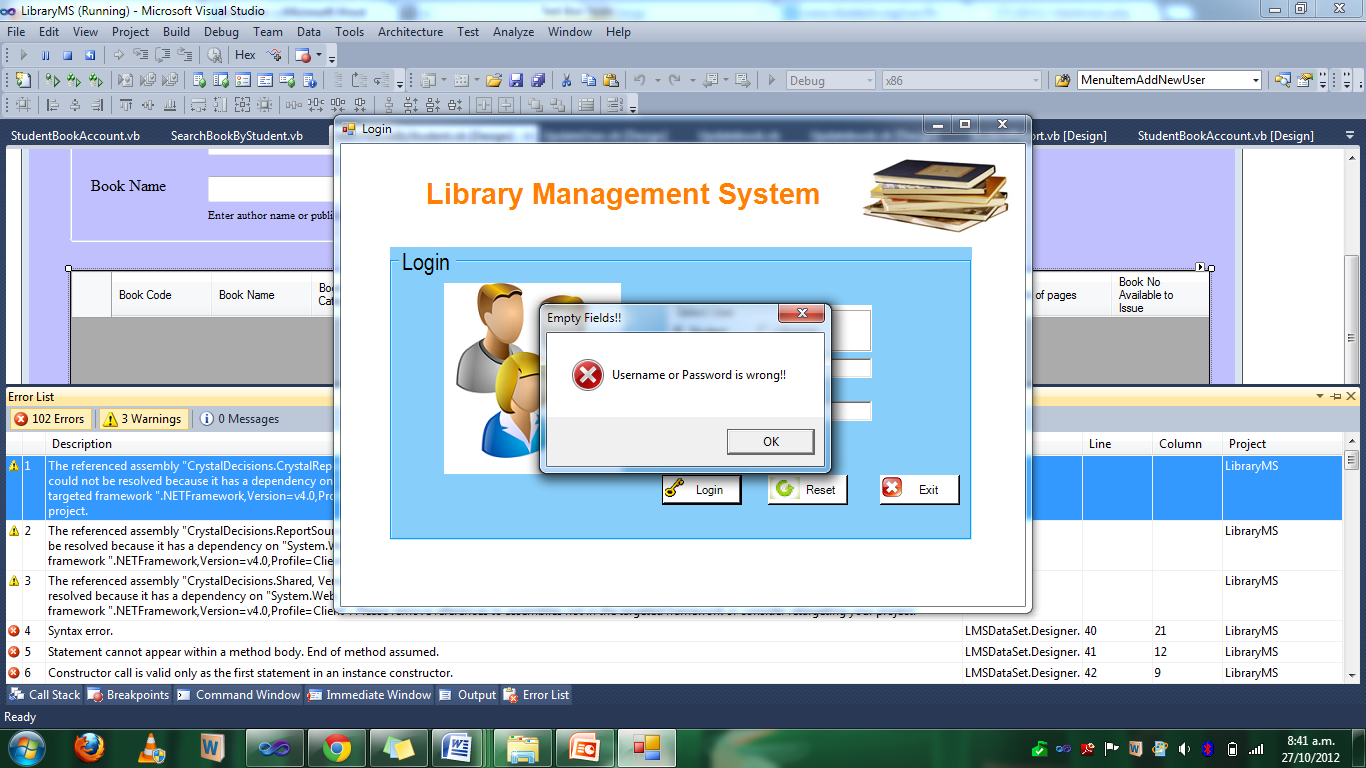
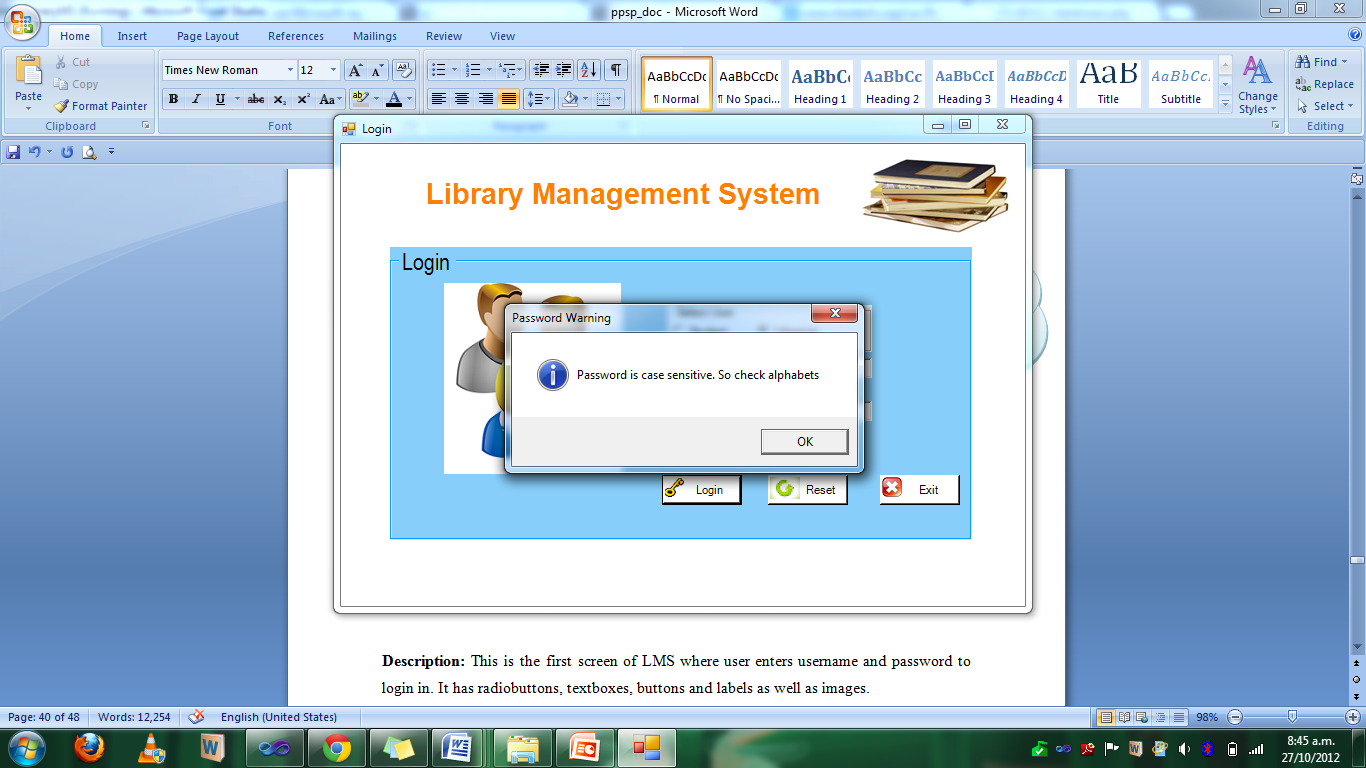
Describes how the software communicates with itself, to systems that interact with it and with humans, the data flow and control flow diagrams provide much the necessary information. The screen designs of LMS are as follows:

**a) Figure 1: Input and output of Login screen**

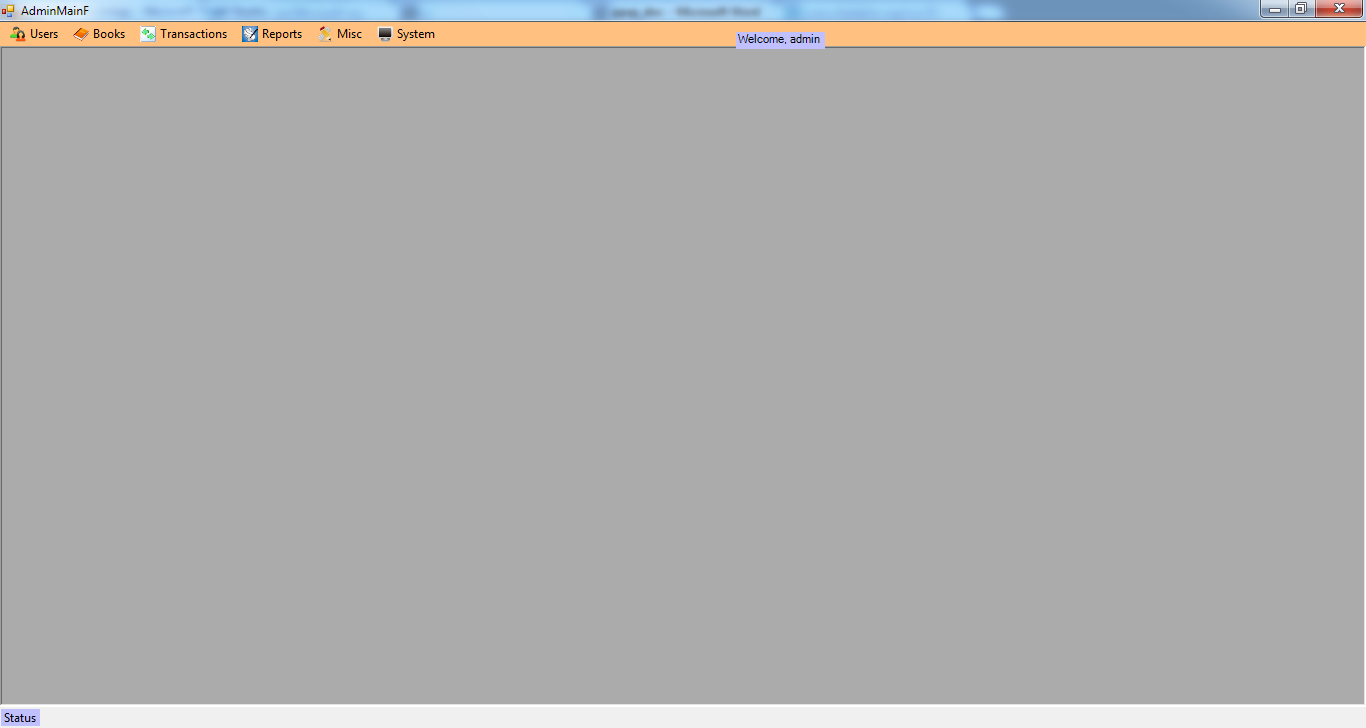
User enters username and password to login into the system. Reset button is used to clear all the fields. Exit is used to exit from the system.



**Description:** This is the first screen of LMS where user enters username and password to login in. It has radiobuttons, textboxes, buttons and labels as well as images. User inputs certain values. If it does not match, the first message below is prompted else if password is entered in Uppercase, the second message is prompted. The user is asked to enter values in correct format.

**b) Figure 2: AdminMainF screen**

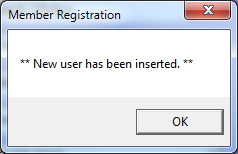


This is the menustrip to select menuitems to perform add, delete, search operations.

**Description:** This is the main screen which consists of menustrip so that librarian can perform any of the add, delete etc operations. Here user does not give any input; just he needs to click on the menuitems to open a form.

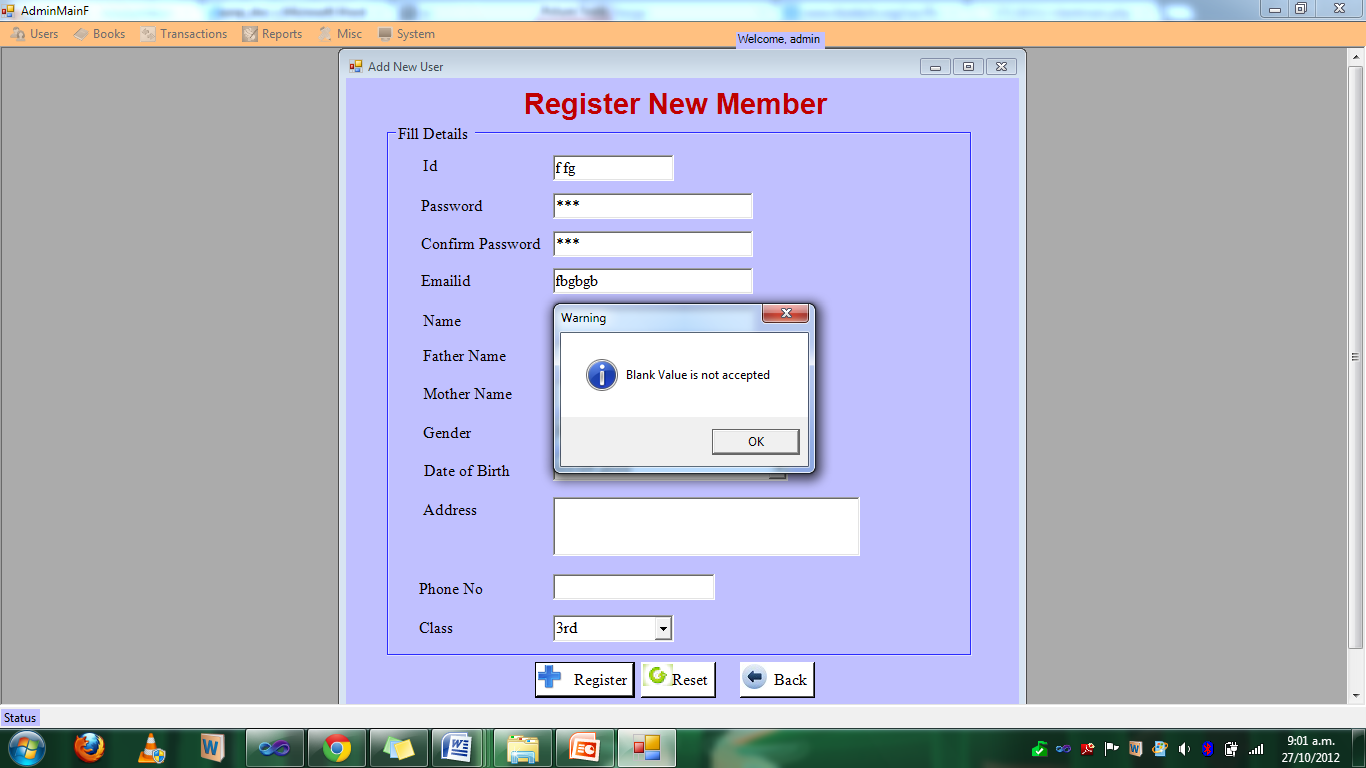
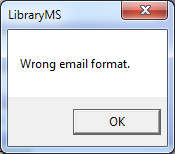
**c) Figure 3: Input and output of Add New User screen**

Here, user librarian enters student details in the specified fields. If the entered values are in correct format, the following message is displayed.

** **

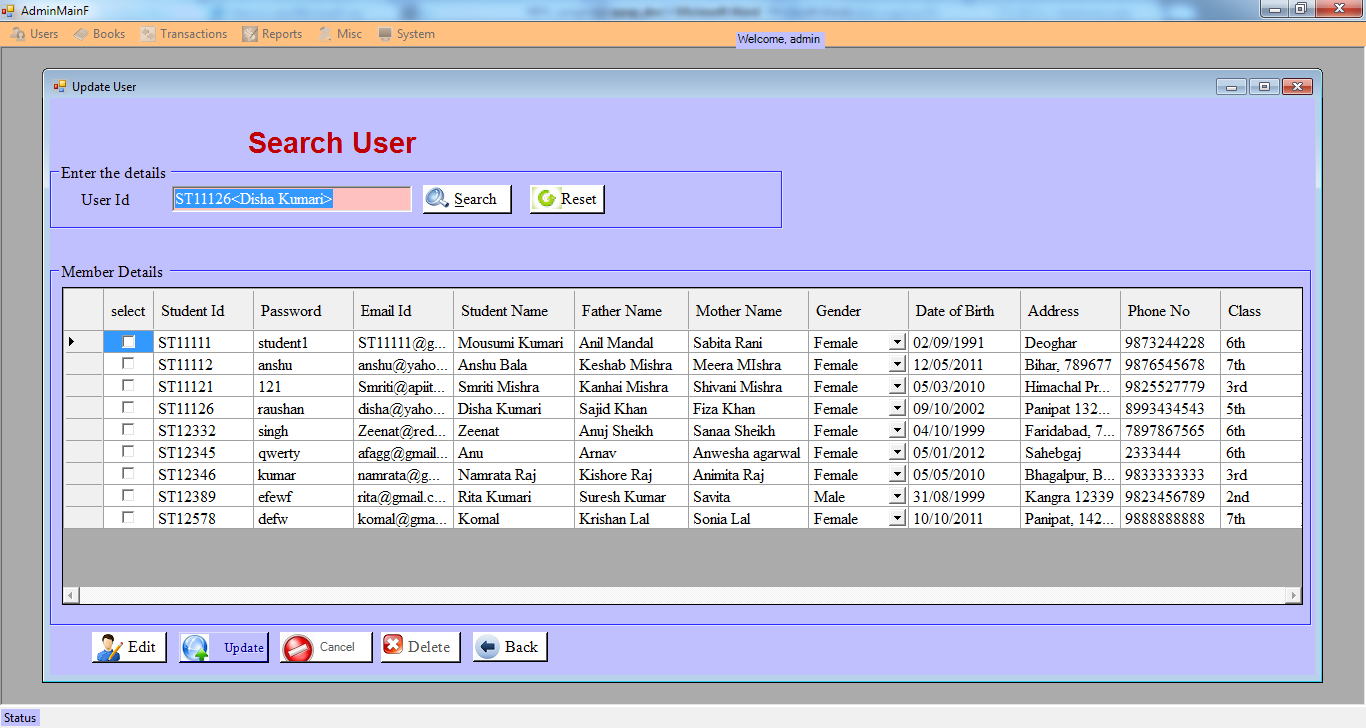
**Description:** Here, librarian can add details of a new student and register him/her. Reset button is used to clear all the fields and back button is used to go to AdminMain form.

If user does not enter values in correct format, error message is displayed. Some examples are shown below, similar type of messages are displayed for other fields also.

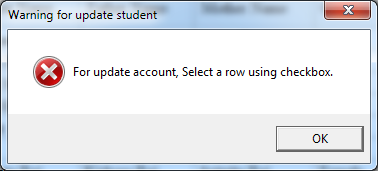
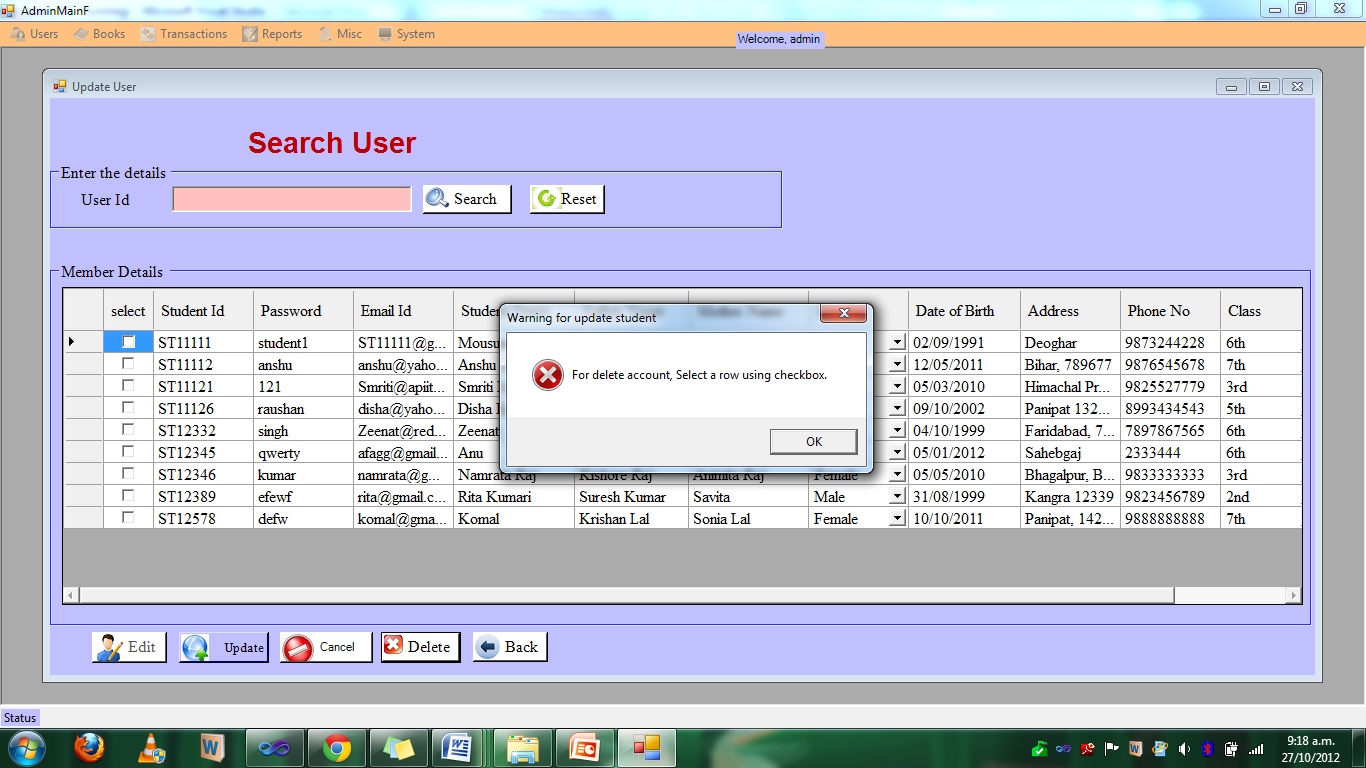
 ** **

**d) Figure 4: Input and output of UpdateUser screen**

Librarian enters the id of the student to be searched and corresponding details appear.

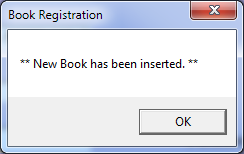
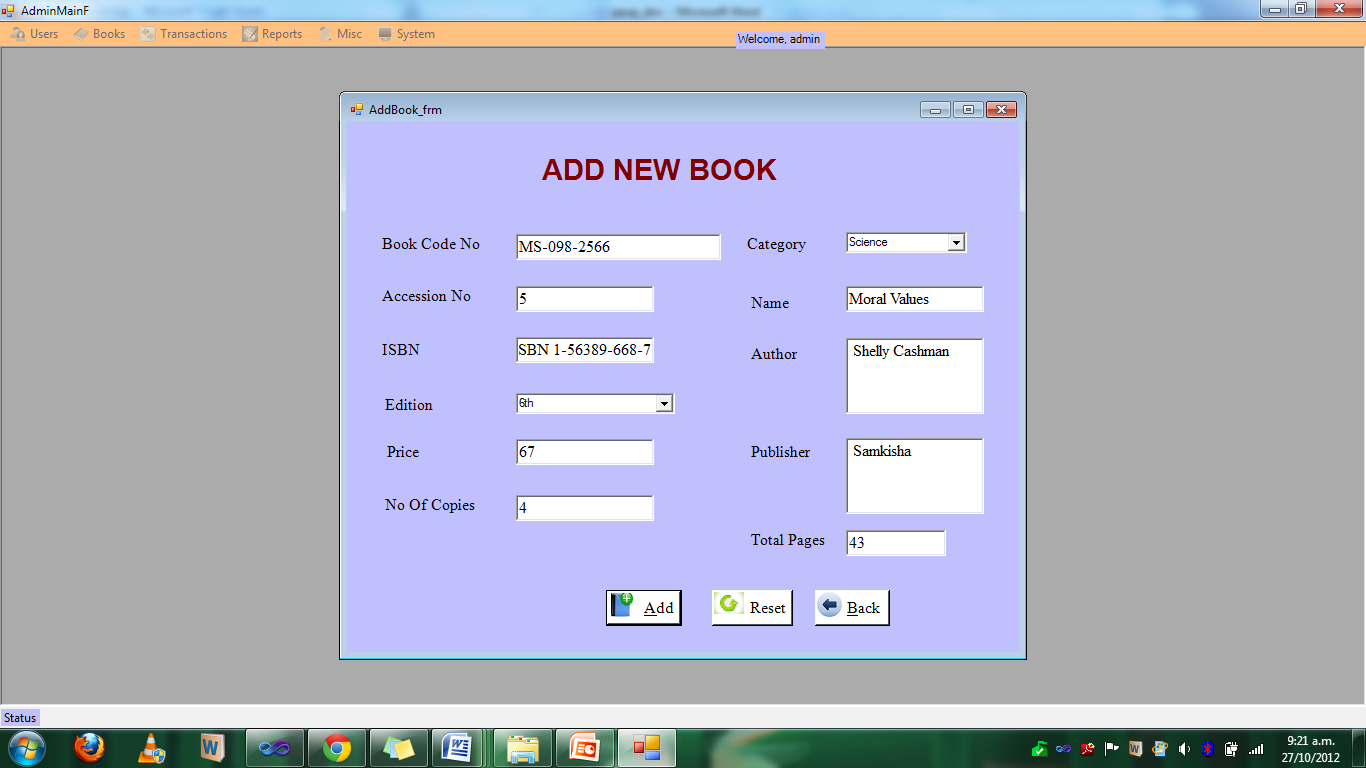
****

**Description:** In this screen, search, edit, update, delete all four operations are performed since edit, update and delete can be done only after searching. If a user, clicks on the update and delete button without clicking on edit button, the following message is prompted else the details of the corresponding user is displayed.

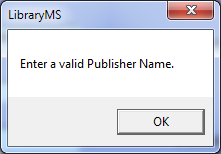
** **

**e) Figure 5: Input and output of AddBook\_frm screen**

Here, user librarian enters book details in the specified fields. If the entered values are in correct format, the following message is displayed.

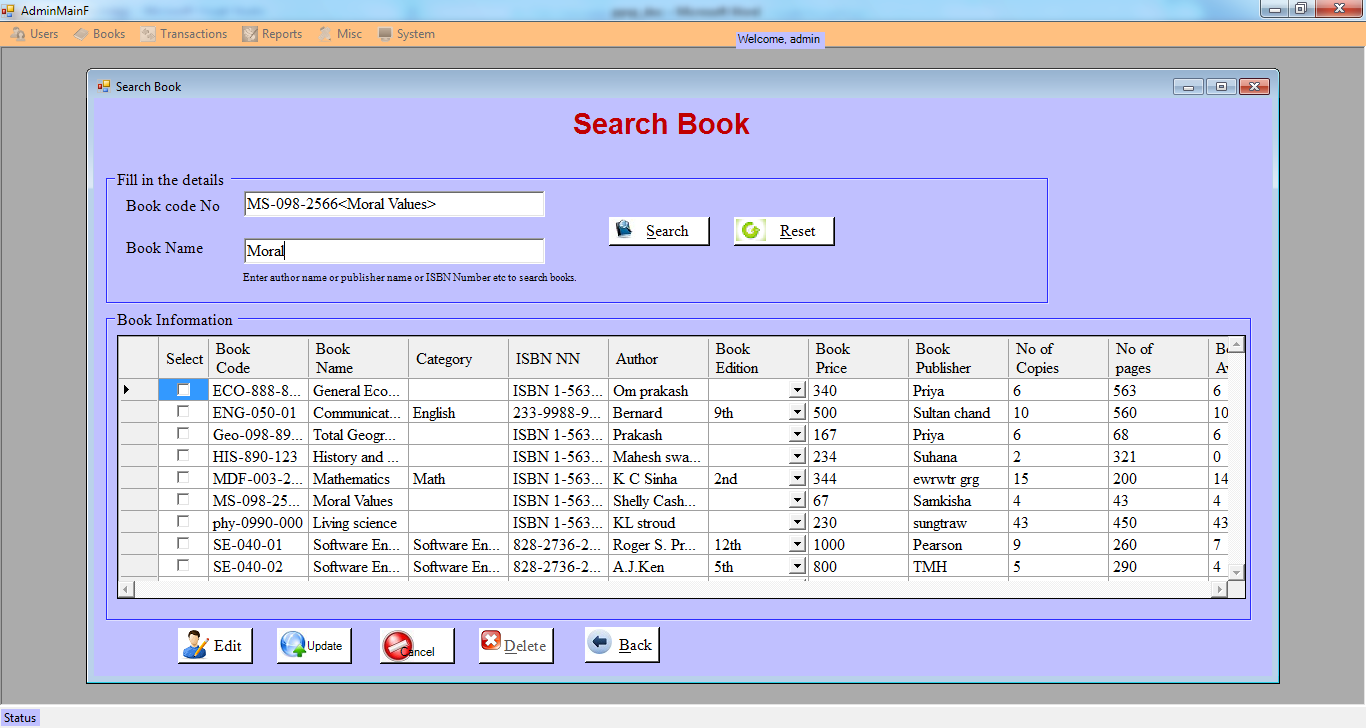


**Description:** Here, librarian can add details of a new book if needed. Reset button is used to clear all the fields and back button is used to go to AdminMain form.If user does not enter values in correct format, error message is displayed. Some examples are shown below, similar type of messages are displayed for other fields also.

** **

Librarian enters the book code No to be searched and corresponding details appear.

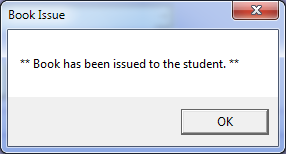
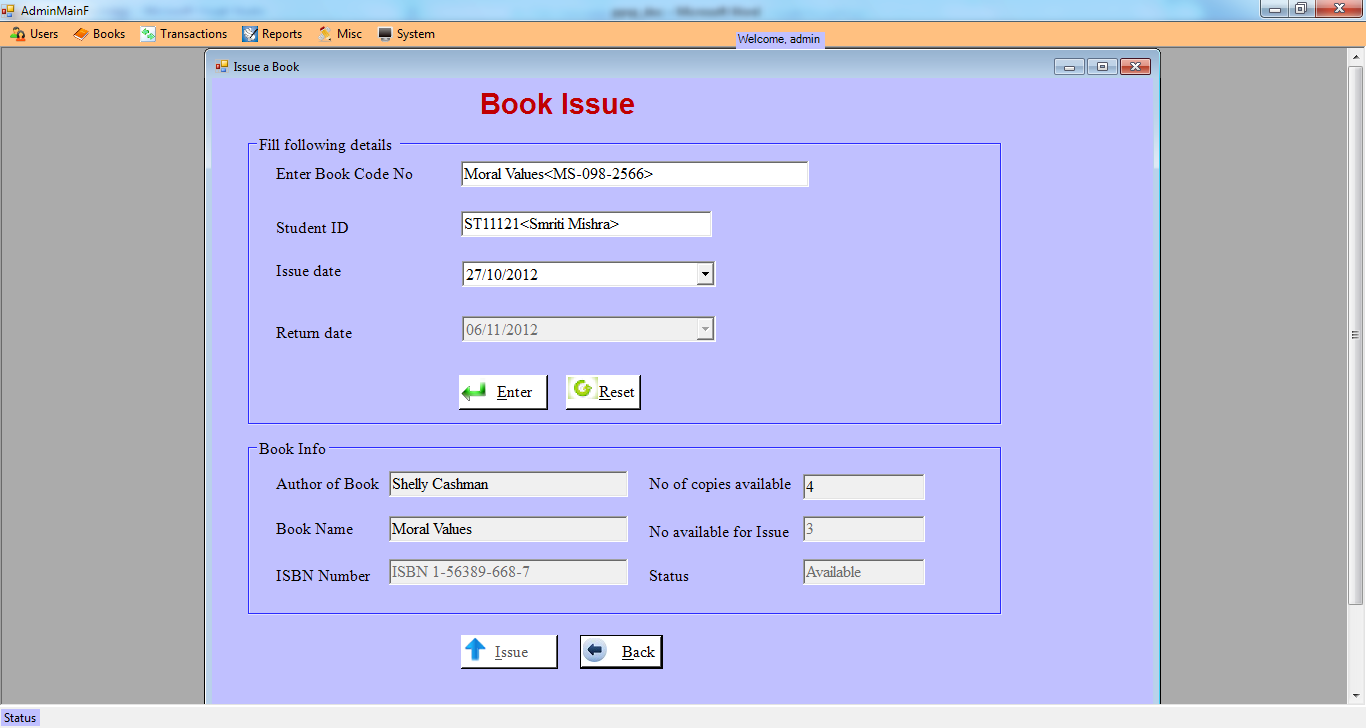
**f) Figure6: Input and output of UpdateBook screen**

****

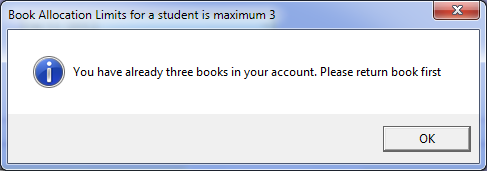
**Description:** In this screen, search, edit, update, delete all four operations are performed since edit, update and delete can be done only after searching. If a user, clicks on the update and delete button without clicking on edit button, the following message is prompted else the details of the corresponding user is displayed. The output screens will be same as updateUser screen’s.

Only three books are issues to a student, if all conditions are fulfilled, book is issued.

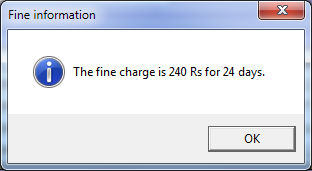
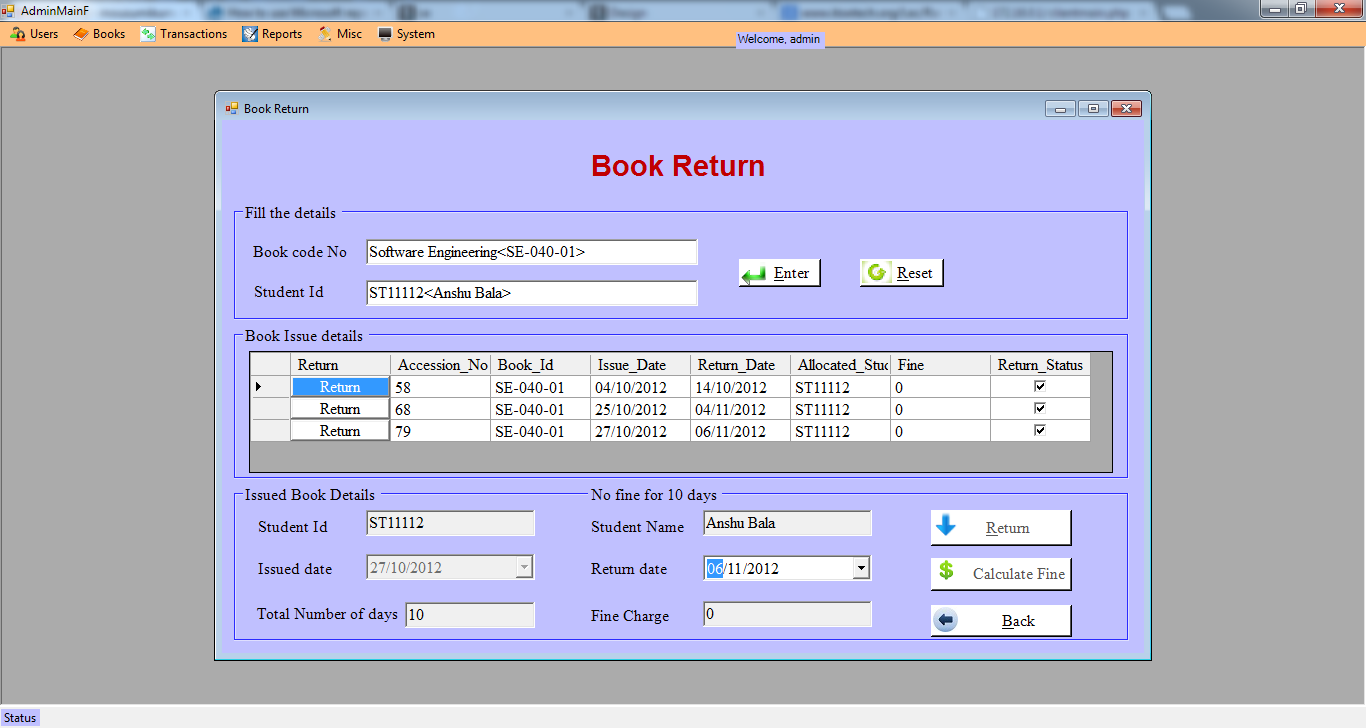
**g) Figure 7: Input and output of Bookissue screen**



**Description:** In this screen, book is issued to students by entering book code or student id. Same copy cannot be issued to more than one student. If still user tries to issue the book, following message is prompted and if same copy is issue to more than one student, issue button is disabled. The screen has enter, reset, issue and back button.

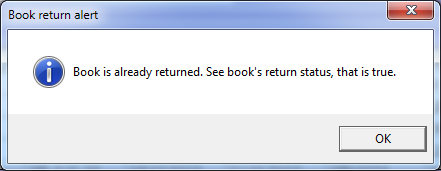
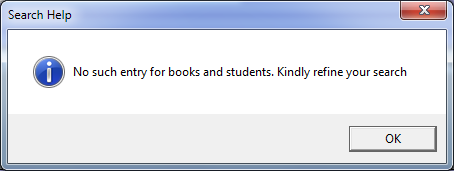


**h) Figure 8: Input and output of Bookreturn screen**

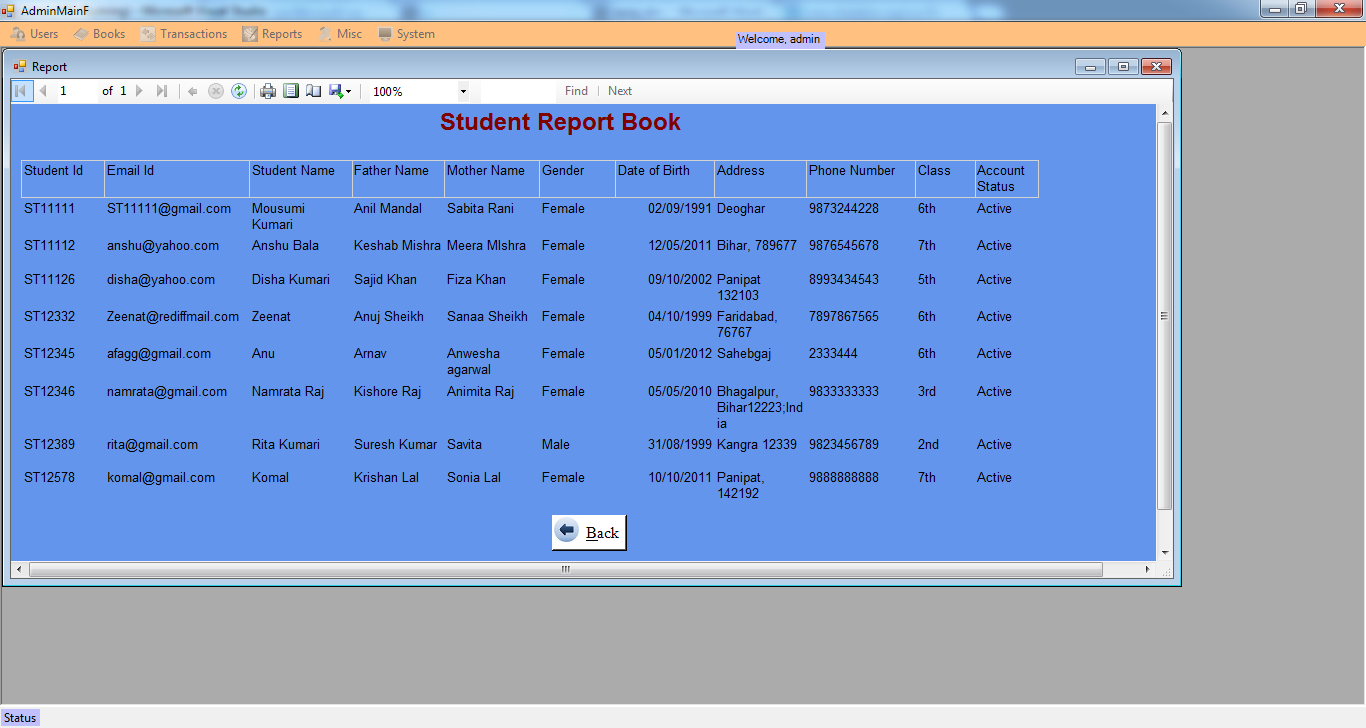
****

On entering book code no and student id, list of books to be returned is displayed. On clicking return, fine is displayed and corresponding issue details is displayed.

**Description:** It has enter, reset, return, fine button. Book is returned after entering book code no or student id and fine is calculated. If a student has not issued the entered book id or if an already book is made to return then, following messages are prompted:

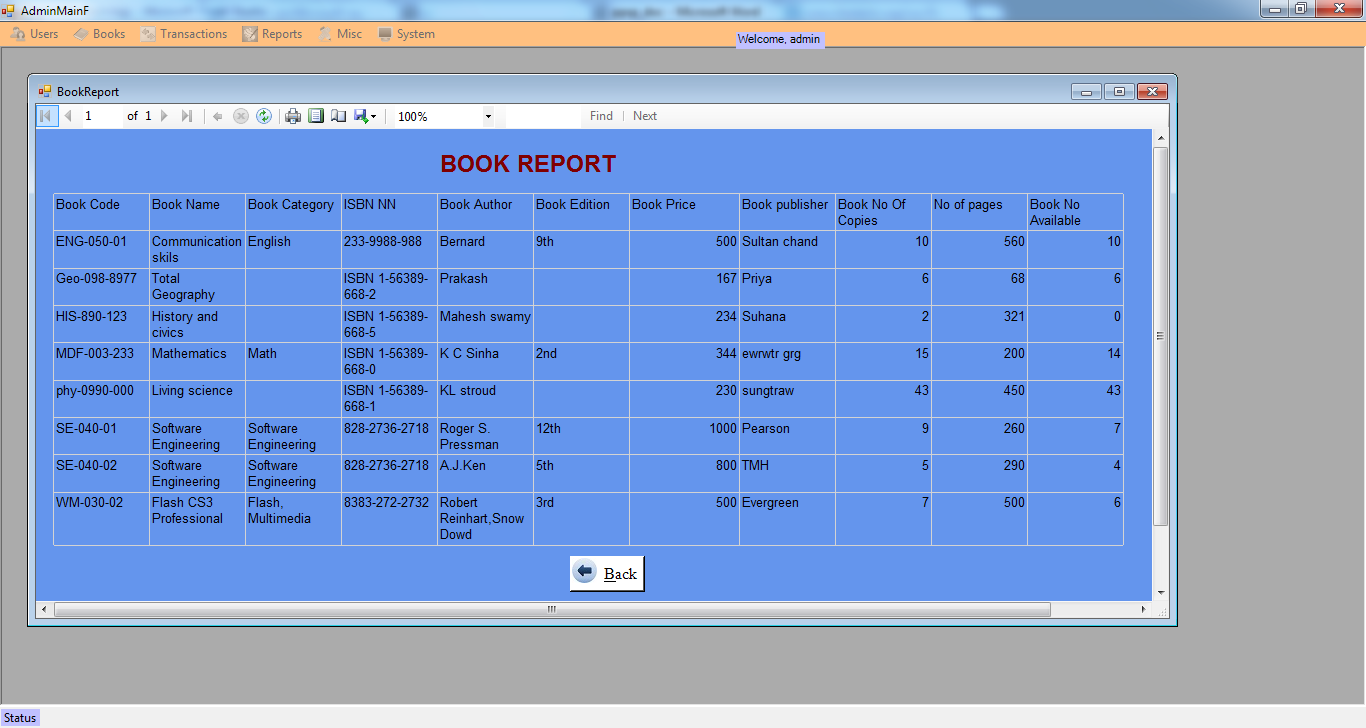


**i) Figure 9: StudentReport screen**

****

**Description:** The student report is generated showing list of students and their information.

**j) Figure 10: BookReport screen**

****

**Description:** It is the book report showing the list of books in the library.

# 19.0. PROGRAMMING ENVIRONMENT

The front end language used to develop LMS is VB.NET and back end language is SQL (SQL server 2008). The platform used is Visual Studio 2010. VB.NET is a powerful window based language which can be deployed easily. It has full object oriented constructs with powerful, flexible and simplified data access. One can code faster and more effectively. A multitude of enhancements to the code editor, including enhanced IntelliSense, smart listing of code for greater readability and a background compiler for real-time notification of syntax errors. On the other hand, SQL server 2008 allows data compression with improved performance. It is executed at backup time to prevent tampering, supports many data types and is flexible. Moreover, it has internal compatibility with VS and .NET framework, can provide 1000 connections at a time with greater DBMS support. It renders improved memory management and faster blocks computations. VS 2010 provides better monitor support, pin watch variables, better searching and large number of controls in toolbox. (<http://www.startvbdotnet.com/dotnet/vbnet.aspx>) (<http://stackoverflow.com/questions/198478/advantages-of-sql-server-2008-over-sql-server-2005>)( <http://www.developerfusion.com/article/84618/new-visual-studio-2010-features/>)

# 20.0. TESTING

Testing is performed to show that a program does what it is intended to do and to discover

program defects before it is put into use.

|  |  |
| --- | --- |
| **Introduction** | This test case checks the functionality of the Librarian login and managing students. |
| **Test Items** | User login, Register Student. |
| **Features to be tested** | Username and password of librarian, student details. |
| **Features not to be tested** | Not Applicable |
| **Approach** | Integration testing. |
| **Test process** | 1. Start application 2. Select librarian from the user type. 3. Enter username and password and click login. 4. Enter all the details of student like name, id, father name, mother name, phone no, email id, password. 5. Click submit   Expected message is shown. |
| **Pass/Fail Criteria** | The test passed in Integration Testing. |
| **Test Deliverables** | Test case report |
| **Plan Approvals** | Name : Ankit Singh Rana Date: 22/10/12 |

|  |  |
| --- | --- |
| **Introduction** | This test case checks whether the Student Registration module is functioning properly or not. |
| **Test Items** | Student Registration |
| **Features to be tested** | Name, id, father name, mother name, phone no, email id, password, gender, date of birth, address, phone number, class |
| **Features not to be tested** | Not Applicable |
| **Approach** | Alpha testing, Beta testing |
| **Test process** | 1. Click add new user. 2. Add new form will be opened 3. Enter all the details like name, id, father name, mother name, phone no, email id, password,etc. 4. Click submit   Expected message is shown. |
| **Pass/Fail Criteria** | The test passed in the alpha as well as beta testing. |
| **Test Deliverables** | Test case report |
| **Plan Approvals** | Name : Mousumi Kumari (Alpha Testing) Date: 23/10/12  Name : Gaurav Kumar (Beta Testing) Date: 23/10/12 |

**APPENDIX A**

**Risk management plan**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Risk Identified | Type of strategy | Strategy description |
| 1 | Loss of data storage devices, the database cannot process as many transactions per second as expected. | Avoidance | Investigate the possibility of buying a higher-performance database. |
| 2 | The risk associated with computer stoppages or any malfunctioning, defects or software corruption. | Avoidance | Proper use of the resources so as to avoid any hardware failures and taking timely backups. |
| 3. | Software is not delivered on time. | Avoidance | Working on project as per schedule, Investigate buying in components, investigate use of a program generator |
| 4 | Inexperienced staff | Avoidance | Alert customers of potentially difficulties and the possibility of the delays, investigate buying-in components. |
| 5 | Staff size | Avoidance | Proper staff-sizing in order to avoid overstaffing or understaffing. |
| 6 | Employee illness | Retention | Reorganize team so that there is more overlap of work and people therefore understand each other’s jobs. |
| 7 | Size underestimate | Retention | Use a different approach to minimize the size. |
| 8 | High work load | Retention | Divide the work into smaller tasks. |
| 9 | Application Complexity | Avoidance | Keep the design simple. |
| 10 | Requirements change | Retention | Derive traceability information to assess requirements change impact, maximise information hiding in the design. |
| 11 | Software corruption | Elimination | Making timely back-up and restoration points. |
| 12 | Hardware failure | Elimination | Invest in good quality hardware components. |

**Introduction to quality**

Quality, simplistically, means that a product should meet its specification. It is composed of 3 C’s namely, customer satisfaction, continuous improvement and continuous loop for solving problems. It is measured in terms of standards and documentation.

**Quality management activities**

Software quality management ensures that the required level of quality is achieved in a software product, defines standards and procedures to be followed and develops a ‘quality culture’ where quality is seen as everyone’s responsibility. The quality management activities are:

* **Quality assurance:** Establish organisational procedures and standards for quality.
* **Quality planning:** Select applicable procedures and standards for a particular project and modify these as required.
* **Quality control:** Ensure that procedures and standards are followed by the software development team.

**Quality assurance plan (QSP)**

Quality assurance is the process of defining how software quality can be achieved and how the organization knows that the software has achieved the desired quality. The two types of standards established as part of the QSP are:

* **Product standards:** They include document standards including documentation and coding standards.
* **Process standards:** They include the definitions of specification, design and validation process and document description of the processes.

|  |  |  |
| --- | --- | --- |
| Factors | Goal (How to remove defects) | Achieved |
| Customer satisfaction | Make the system usable and reliable. | 60% |
| Unable to make changes in database. | The checkbox of prevent changes when table recreation was checked. | 80% |
| Conversion error in datetime picker | Changed the format to dd/mm/yyyy | 90% |
| Argument exception | Exchanged the positions of search and insert query | 70% |
| Crystal report generation problem | Installed crystal report software | 75% |
| Updation in datagridview table | Researched for datagrid view properties | 87% |
| Days were calculated in negative | Changed the propery of Datetimepicker | 70% |

**Process quality**

Process quality influences the quality characteristics of software products, which in turn affect quality-in-use as perceived by the customer. Software quality management and software engineering process quality have a direct bearing on the quality of the software product. (<http://www.computer.org/portal/web/swebok/html/ch11>)

|  |  |  |
| --- | --- | --- |
| Name of process | Description | Document standard |
| Planning | Identify the problems, determine the scope and objectives of the project and propose a solution, select a methodology. Do project planning, cost estimation, project management and project scheduling. | Gantt chart, PERT chart, project organization chart, work breakdown structure, staffing plan. |
| Analysis | Find out what system will do. Use data gathering methods to find out requirements of users. Do analysis modelling (data, process and behavior modelling) | Context diagram, DFD level0, level1, level 2, ERD, ELH, data dictionary, structured English, decision tree. |
| Design | Map analysis model to design model, database, and procedural and screen designs. | Data, architecture, interface and procedural design. |
| Construction | Build and test the system. | Test plan and test cases. |
| Implementation and evaluation | Training, conversion of files, installation. See if intended users are using the system. | - |
| Maintenance | Use and maintain the system. | - |

**Process improvement**

Process improvement means understanding existing processes and changing these processes to increase product quality and/or reduce costs and development time. It is a cyclical activity. It involves three principal stages: process measurement, process analysis and process change.

|  |  |  |
| --- | --- | --- |
| Name of process | Estimated no. of defects | Actually removed |
| Planning | 25 | 19 |
| Analysis | 20 | 15 |
| Design | 23 | 17 |
| Construction | 35 | 27 |
| Implementation and evaluation | 10 | 5 |
| Maintenance | 5 | 3 |

**REFERENCE**

Sommerville Ian (1988), *Software Engineering*,7th ed.South Asia: Pearson Education Ltd.

Visual Basic .NET Features. 2012. Visual Basic .NET Features. [ONLINE] Available at: http://www.startvbdotnet.com/dotnet/vbnet.aspx. [Accessed 27 October 2012].

Advantages of SQL Server 2008 over SQL Server 2005? - Stack Overflow. [ONLINE] Available at: http://stackoverflow.com/questions/198478/advantages-of-sql-server-2008-over-sql-server-2005. [Accessed 27 October 2012].

Expert Project Management - Project Management State of the Art - 2004: Part 1. [ONLINE] Available at: http://www.maxwideman.com/guests/stateofart/characteristics.htm. [Accessed 27 October 2012].