**INTRODUCTION**

In the world of computers everything becomes fast. Computer plays a wide role in any organization. Nobody wants to spend maximum part of his precious time in Searching. The project “**LIBRARY MANAGEMENT SYSTEM**” is time saving and informatics application for every employee. The main and fore most objective of this project is to provide facility for maintaining the details of LIBRARY.

Database is an organized collection of related information. The data is organized in some format, it becomes information refers to organized data and group of organized and related information. A DBMS (Data Base Management System) is a program or collection of Programs that allows number of users to access and modify data in a database. It is a software package used to build and manage the data. The DBMS interface’s tools to perform data management functions are creating tables, entering and editing data, viewing data, sorting records, querying the database and generating reports.

The biggest asset of DBMS is its ability to provide quick access and retrieval from large databases. Because database files can grow extremely large, recalling data quickly is not a trivial matter .A DBMS, especially when it is running on powerful hardware, can find any speck of data in an enormous database in minutes- sometimes even in seconds or in fraction of seconds. Many different DBMS programs are available. Enterprise level products, such as Oracle,MS ACCESS and SQL Server are designed to manage large corperate or special purpose database system.

**PROBLEM DEFINITION**

**PROFILE OF THE PROBLEM**

INTRODUCTION:

The project entitled “LIBRARY MANAGEMENT SYSTEM” is created for managing and meet out the requirements of a library. Specially, in this modern time, the computer is the most important that everyone wants to apply in their general lives and business ventures. So, in this point of view this project may helpful to do the front office job in a library easily and in a very few time. The Graphical User Interface and Fast Report Generation technique are the major aspects in the project. With the minimized inputs one can easily handle the front office job in a library using this application. This project will be capable of managing the book information i.e. issued book, returned book, book information.

PURPOSE:

The increasing courage of technology and computers can made any business venture to feel the requirement of such an application. As I said the application is capable of managing the book information i.e. issued book, returned book, book information, it make the front office job very easy. It will require less efforts, less paper work and having less error prone. With a central database for every front office activity, the job will be easy without any hurdle.

SCOPE:

Future scopes for this project are bright. The application will comprise of information regarding the issued book, returned book, , book information. This application can be used in any organization dealing with the issue and return of books in library because each organization is interested in computerized management of its working, which makes easy to analyze performance of work. It is useful to find out the machines status or their monitoring at management level, as well as a whole whenever is required immediately without waiting as all data will be computerized and reports well be back away.This application is easily extensible and upgradeable according to organization need little or no change.

**FEASIBILITY STUDY**

**FEASIBILITY ANALYSIS**

FEASIBILITY:

Feasibility analysis is a test of the system proposal according to its workability, impact on organization, ability to meet user needs and effective use of resources. The objective of the feasibility analysis is not to solve the problem but to acquire sense of its scope. An initial investigation culminates in proposal that determines whether an alternative system is feasible. A proposal summarizing the thinking of the analyst is presented to the user for review.

Steps in feasibility analysis:

* Form a project team and appoint a project leader.
* Prepare system flowcharts.
* Enumerate potential proposed systems.
* Define and identify characteristics of proposed system.
* Determine and evaluate performance and cost effectiveness ofeach proposed system.
* Weight system performance and cost data.
* Select the best proposed system.
* Prepare and report final project directive to management.

TYPES OF FEASIBILITY:

For approving the development of proposed system, three major aspects in the feasibility analysis are considered. These are:

* Economic Feasibility
* Technical Feasibility
* Operational Feasibility

ECONOMIC FEASIBILITY:

To analyze the economic feasibility, the procedure known as Cost Benefit analysis is used. This procedure helps to determine the benefits and saving that are expected from a proposed system and compare it with the cost. Since the benefits outweigh the costs so we made the decision to design and implement the system. Since the existing system is manual so proposed system reduces manpower due to which it leads to lot of savings.

TECHNICAL FEASIBILITY:

This feasibility centers on the existing computer system hardware and software and to what extent they support the proposed system. It includes:

* Feasibility to produce outputs in a given time.
* Response time under certain conditions.
* Ability to process a certain volume of transaction at a particular speed.

For our project, hardware requirements were limited and for software we needed JAVA as front end and MS-Access as back end. Since the budget was not the serious constraint, the project was feasible. Since the existing system is manual so there was no problem to install new software and hardware.

OPERATIONAL FEASIBILITY:

People are inherently resistant to change, and computers have been known to facilitate a change. As estimation is made to how strong a reaction, the employees are likely to have, towards the development of an automated system. Dramatically it is found that users are very keen to use this new System as this will help them in reducing their work load. For adapting to this system they have already started taking training to learn computers. So the proposed system is Operational Feasible too.

**SOFTWARE SYSTEM ANALYSIS**

**SOFTWARE SYSTEM ANALYSIS**

Nowadays, quality has become an important factor to be considered while developing software. This is due to the fact that users are interested in quality software, which is according to their requirements and is delivered within a specified time. Furthermore, users require software which is maintainable and minimizes the time and cost for correcting problems in it. The software should fulfill the required functionality as specified by the user and there should be minimum errors according to those requirements. The quality of software depends on various characteristics, such as correctness, reliability, efficiency, portability etc. These are used as a checklist to implement quality in software. Thus analyzing the software system becomes necessary because of quality and standard. Here are some issues related to this topic.

RELIABILITY:

It refers to the ability of the software to perform required function under given conditions for a specified period. It is the process of optimizing the reliability of software through program that emphasizes software error prevention, fault detection and removal, and the use of measurements to maximize reliability in light of project constraints like resources, schedule and performance.

AVAILABILITY:

The system will be available to users, 24 hours a day, all through the year expert at times while the database is brought down i.e. for preventive/corrective maintenance or backup etc.

MAINTAINABILITY:

The system should be developed using high technologies for all stages of design and development, making the system documentation like software requirements specification (SRS), design document deliverables. The use of these processes and techniques will ensure the delivery of a system with high maintainability. Maintainability refers to the ease with which a software system can be modified to add capabilities, to improve system performance or correct errors.

PORTABILITY:

It refers to the ease with which software developers can transfer software from one platform to another, without or with minimum changes. It refers to the ability of software to function properly on different hardware and software platforms without making any changes in it.

EFFICIENCY:

The efficiency of system refers to the ability of the software to use system resources in the most effective and efficient manner.

**DATAFLOW DIAGRAM**

DATA FLOW-DIAGRAMS:

It is a way of expressing system requirement in a graphical form. This leads to a modular design. It is also known as bubble chart, has the purpose of clarifying system requirements and identifying major transformations that will become program in system design. So it is the starting point of the design phase that functionally decomposes the requirement specifications down to the lowest level of details. A DFD consist of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flows in the system. Here is an introduction to DFD Symbols:

PROCESS:

A process or also known as function is represented using circle. The symbol is called as bubbles or a process. Bubbles are annotated with the names of the function.

SOURCE OR DESTINATION OF DATA:

Source or destination of data is those physical entities external to the software system, which interact with the system. By inputting data to the system or consuming the data produced by the system.

DATA FLOW:

A directed arrow is used as a data flow symbol. A data flow represents the data flow occurring between two processes, or between an external entity and a process. Data flow is annotated with the corresponding data name.

DATA STORAGE:

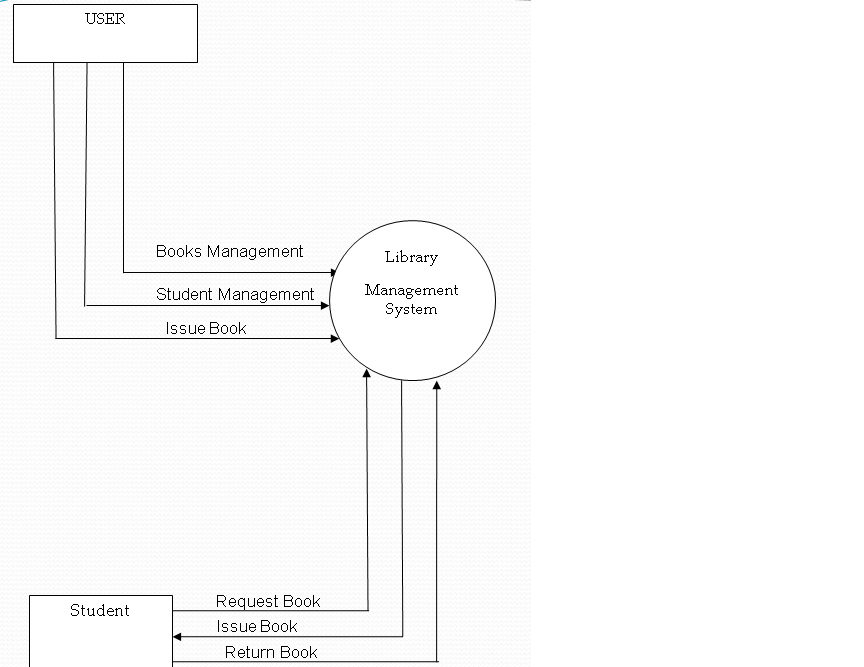
A data store is like a warehouse for data. It is represented using two parallel lines .Each data store is connected to a process using data flow symbol. Processes may retrieve or store data from a file. If arrow points to the file it indicates operation of writing into the file. If it points away from the file, it indicates operation of reading from the files

.

OUTPUT:

The output symbol, is used when a hard copy is produced and user of the copies cannot be clearly specified or there are several users of the out.

**0-Level DFD:**



**1st – Level DFD:**

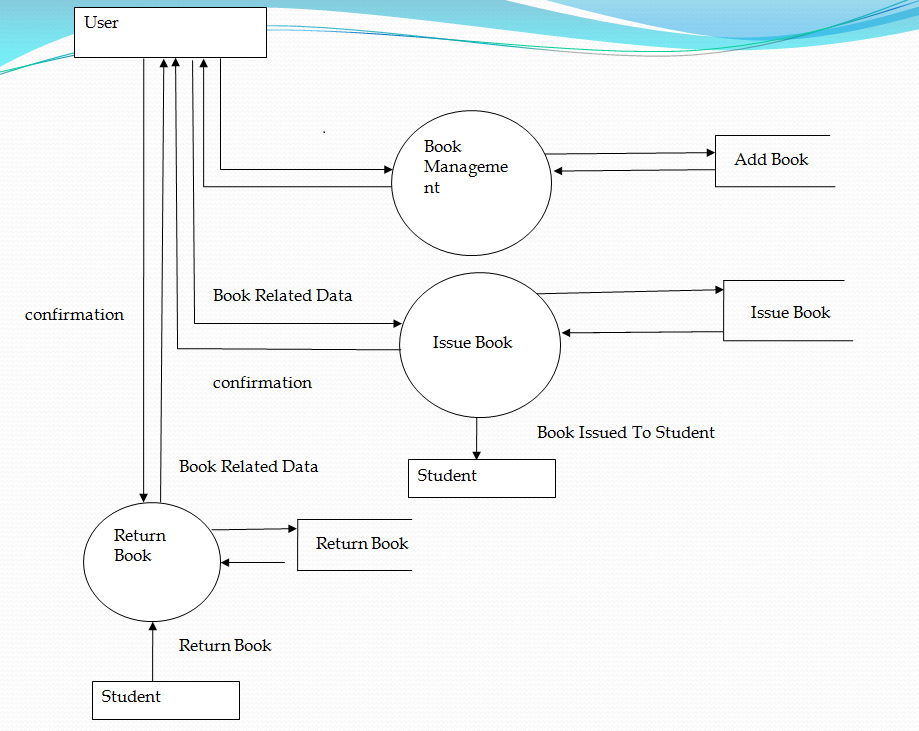


Fig. 1st level DFD

**ENTITY RELATIONSHIP DIAGRAMS**

**ER-DIAGRAMS**

ER diagram defines all the data that are entered, stored, transformed and produced within an application Entity relationship diagram also known as object/relationship diagram shown the relationship between entities. It includes data objects, attributes and relationship. Symbols used in constructing ER Diagrams:

ATTRIBUTE:

Various data items that describe an entity are known as attribute.

ENTITY SET:

An entity is a thing which can be easily identified .It may be object, person, place, things or events of interest and about which data are captured, stored or processed.

RELATIONSHIP:

An association of several attributes in an Entity-Model is called relationship. This connects entities and represents meaningful dependencies between them.

RELATIONSHIP

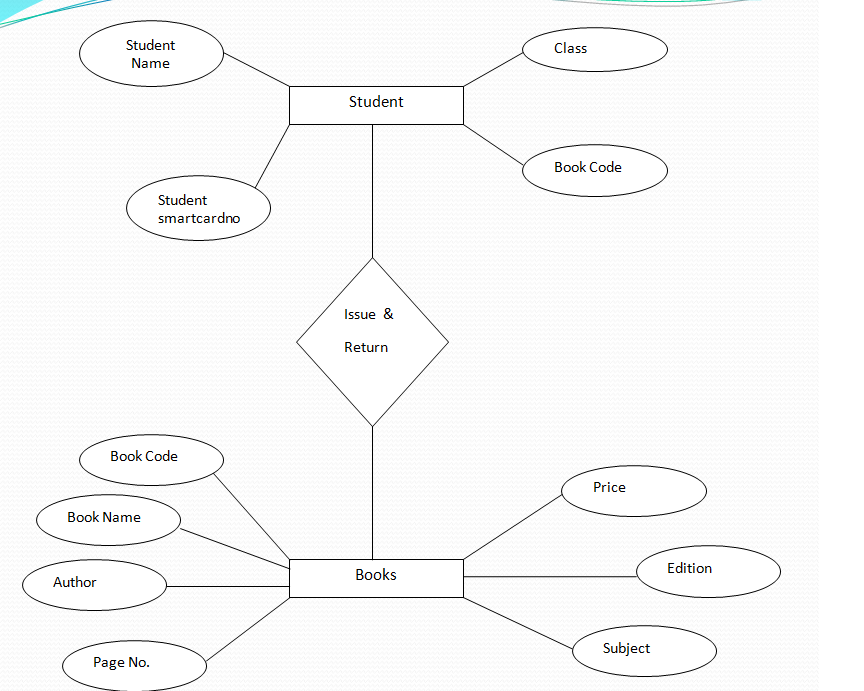
There are three types of relationships which exist among entities. They are

1. One-to-One

2. One-to-Many

3. Many-to-Many

**Entity Relationship Diagram:**

****

**SYSTEM DESIGN**

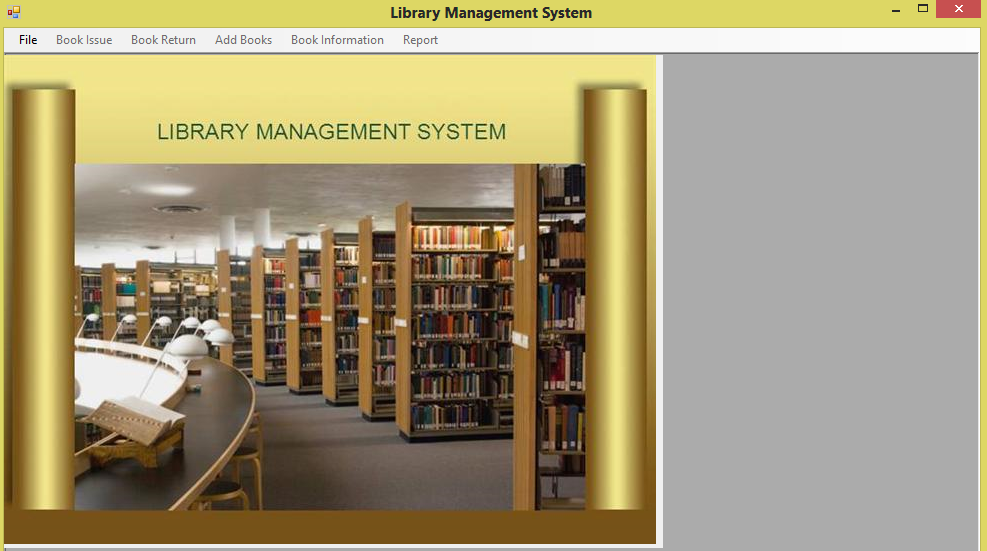
**SYSTEM DESIGN**

Here is the module description for Library management system:

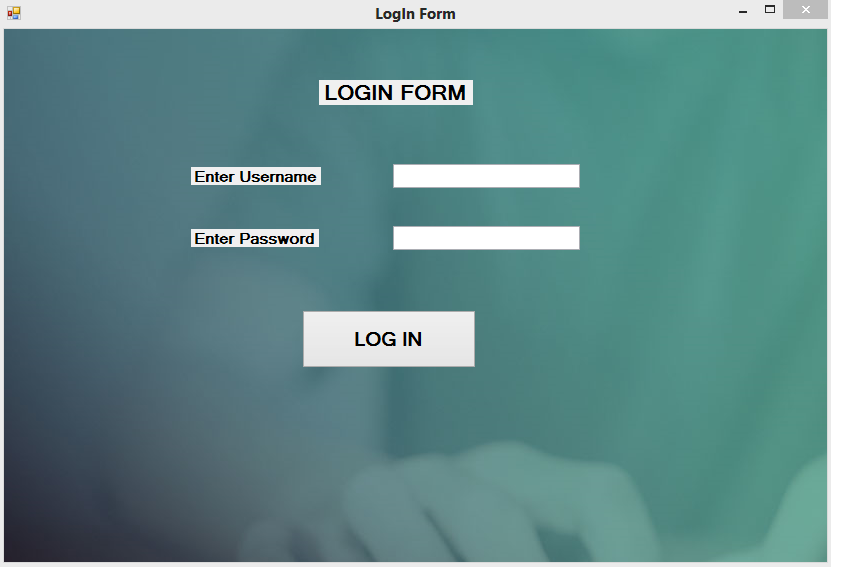
* MDI Form: This is the Multiple Document Interface for the application. This module contains links to open further forms/modules.
* Issue Book: This module comparises of different books issued to the students.
* Return Book: This module is capable of keeping details regarding returned Book.
* Book Information: This form is capable of searching different books available in the library.
* Add Book: This form is capable of keeping the record of different books available in the library.

**SCREENSHOTS**

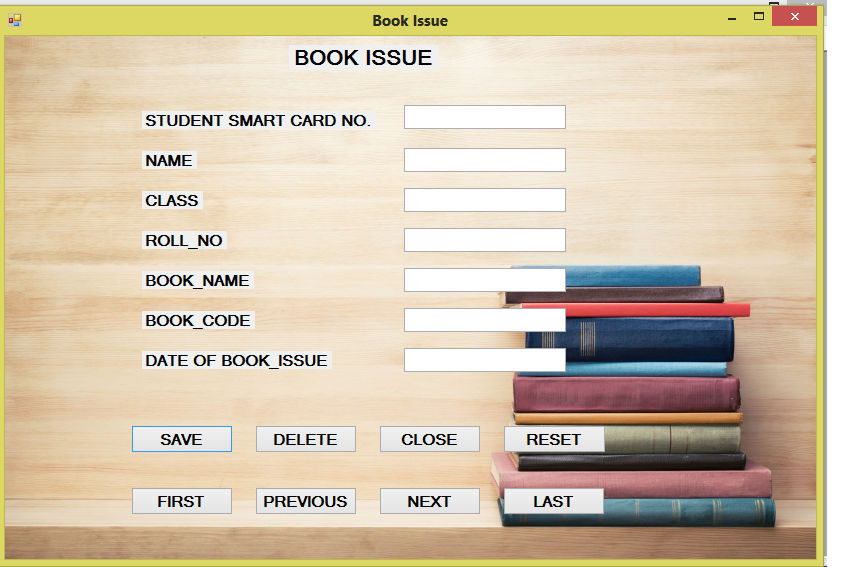
**PROJECT DESIGN**



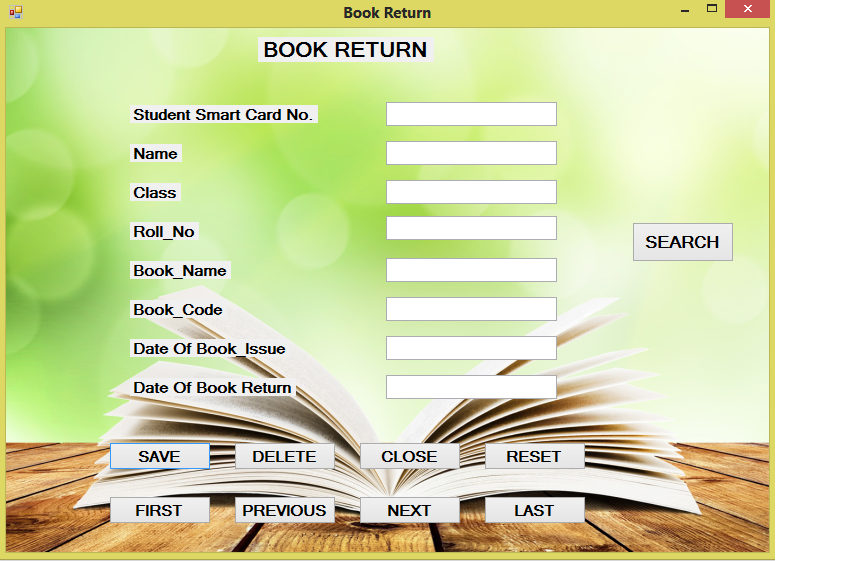
Log In Form



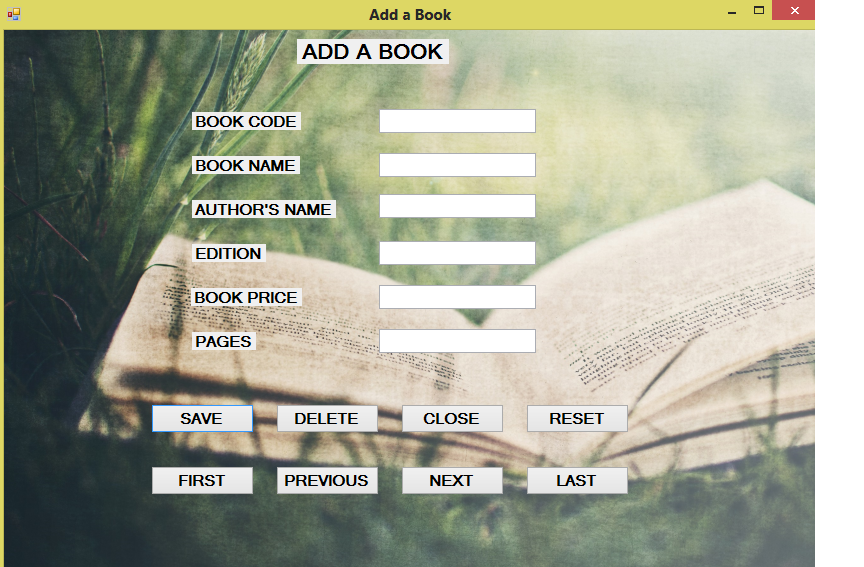
**ISSUE BOOK**



**RETURN BOOK**

****

**ADD BOOK**



**BOOK INFORMATION**



**DATABASE DESIGN**

**TABLE DESIGN**

Table Name: Add Book

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Constraints |
| Book Name | **Text** |  |
| Author | Text |  |
| Book Code | Number | Primary Key |
| Book Price | Number |  |
| Purchase Date | Date / Time |  |
| Pages | Number |  |
| Edition | Text |  |
|  |  |  |

Table Name: Issue Book

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Constraints |
| Student Smart Card Number | Number | Primary Key |
| Student Name | Text |  |
| Class | Text |  |
| Book Name | Text |  |
| Book Number | Number |  |
| Author Name | Text |  |
| Subject | Text |  |
| Date Of Issue | Date/Time |  |
|  |  |  |

Table Name: Return Book

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Constraints |
| Student Smart Card Number | Number | Primary Key |
| Student Name | Text |  |
| Class | Text |  |
| Book Name | Text |  |
| Book Number | Number |  |
| Author Name | Text |  |
| Date Of Issue | Date/Time |  |
| Date Of Return | Date/Time |  |
|  |  |  |

**SYSTEM ANALYSIS**

**SYSTEM ANALYSIS**

Developing software is a difficult undertaking. Lots of things could go wrong and frankly often things do go wrong. Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty and effectively. The risk analysis and management of the project entitled “ASPIRE – Easy Accounting Solution for Automobiles Business” includes the following points:

Scope & purpose of the document: This document records all the work performed as part of the risk analysis .it will cover all the major risks involved in the development of the system under consideration. It will also allocate of those risks:

Overview of major risks: The major risk involved in our system is of two types – project risks & Technical risks, Project risks and Scheduled risks. The project schedule can go beyond the decided date because of variety of reasons ranging from on line job problems, examinations to reduced motivation towards the project by the project team.

Personal risks: The communicational gap between the project team and the project management may affect the motivational level of the members towards the project. Also the intra-relationship between the various team members of the project may hamper the progress of the project in a timely manner. Further the inability of the management to successfully manage their resource well may affect the staffing and organization of the team members to the hilt.

Requirement problems: Unclear definition and description of the requirements may affect the project’s success.

Other Risks: The project may also be affected by other factors such as the project complexity and size & degree structural uncertainty.

Technical risks: Frequent change of the requirements from the user may result in conflicting design or confusion in the mind of the designer leading to an unstable system design.

Technical obsolescence risks: There is a general tendency by both the user and the developer in choosing a technology which has been for quite some time in the market for obvious reasons .but may end up using a technology which is already obsolete or on the verge of its obsolescence. This may result in very quick change in system again.

Implementation risks: Because of non-conversance of the technology being used by the project team which was used just on the insistence of the user, the system being developed may be of compromised quality.

**SYSTEM ENVIRONMENT**

**SYSTEM ENVIRONMENT**

HARDWARE REQUIREMENTS:

* Processor: P4 or above (recommended)
* Ram: 256 MB or above (recommended)
* Display: Any VGA with 1024 X 728 or above resolution
* Printer: Standard office printer can be used (Laserjet recommended)

SOFTWARE REQUIREMENTS:

* Operating system: Windows XP SP-2 or above
* Front End: Visual Studio 2005 or above
* Programming Language: VB.Net
* Back End: MS-Access 2003 or above

**PROJECT TESTING**

**PROJECT TESTING**

What is testing?

Testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. For example, in software development, product objectives are sometimes tested by product user representatives. When the design is complete, coding follows and the finished code is then tested at the unit or module level by each programmer; at the component level by the group of programmers involved; and at the system level when all components are combined together.

Testing Techniques:

There are following techniques of project testing:

* Black Box Testing
* White Box Testing
* Gray Box Testing

TESTING TYPES:

There are following techniques of project testing:

* Unit Testing
* Integration Testing
* System testing or Functional testing
* Acceptance Testing

BLACK BOX TESTING:

It takes an external perspective of the test object to derive test cases. These tests can be functional or non functional, though usually functional. The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object's internal structure. This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance. The higher the level, and hence the bigger and more complex the box, the more one is forced to use black box testing to simplify. While this method can uncover unimplemented parts of the specification, one cannot be sure that all existent paths are tested.

BLACK BOX TESTING:

An internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. In electrical hardware testing, every node in a circuit may be probed and measured; an example is in-circuit testing (ICT). Since the tests are based on the actual implementation, if the implementation changes, the tests probably will need to change, too. For example ICT needs updates if component values change, and needs modified/new fixture if the circuit changes. This adds financial resistance to the change process, thus buggy products may stay buggy. Automated optical inspection (AOI) offers similar component level correctness checking without the cost of ICT fixtures; however changes still require test updates. While white box testing is applicable at the unit, integration and system levels of paths within a unit, it can also test paths between units during integration, and between subsystems during a system level test. Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed.

GRAY BOX TESTING:

Gray box testing technique is often defined as a mixture of black box testing and white box testing techniques. This technique is similar to white box testing in the aspect that it requires limited knowledge of the internal details of the system under test, in order to design the test cases. However like the black box testing, the system is treated as a black box and is tested from the outside. In gray box testing, test design is developed using the information such as state based models or the architecture diagrams of the system to be tested. In addition the inputs are applied to the system and the outputs are observed using the black box approach. The main objective of gray box testing is to find those defects that occur due to poor design or poor implementation of the system.

TESTING CATEGORIES:

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

UNIT TESTING:

In computer programming, unit testing is a software verification and validation method where the programmer gains confidence that individual units of source code are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class. Unit testing can be done by something as simple as stepping through code in a debugger; modern applications include the use of a test framework such as x Unit. Ideally, each test case is independent from the others. Substitutes like method stubs, mock objects, fakes and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its requirements and behaves as intended. Its implementation can vary from being very manual (pencil and paper) to being formalized as part of build automation

.

INTEGRATION TESTING:

Integration testing (sometimes called Integration and Testing, abbreviated I&T) is the activity of software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

SYTEM TESTING OR FUNCTIONAL TESTING:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limiting type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

ACCEPTANCE TESTING:

In engineering and its various sub disciplines, acceptance testing is black-box testing performed on a system (e.g. software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery. It is also known as functional testing, black-box testing, release acceptance, QA testing, application testing, confidence testing, final testing, validation testing, or factory acceptance testing. In software development, acceptance testing by the system provider is often distinguished from acceptance testing by the customer (the user or client) prior to accepting transfer of ownership. In such environments, acceptance testing performed by the customer is known as user acceptance testing (UAT). This is also known as end-user testing, site (acceptance) testing, or field (acceptance) testing.

**PROJECT IMPLEMENTION**

**PROJECT IMPLEMENTATION**

Successful implementation is the crucial phase in the system life cycle of new system design. Implementation includes all those activities that take place to correct from old system to the new system. The new system may be completely new, replacing an existing manual or automated system or it may be major modification to an existing system. In other case, proper implementation become necessary so that a reliable system based on the requirements of the organization can be provided. During the implementation stage, a live demo is undertaken and made in front of the users. The various features provided in the system were discussed during implementation. Doubts and clarifications were cleared immediately. It, basically, involves installing the hardware and software, training the users before the tool is on and running.

Approach Followed:

The approach followed to implement the project is the top down approach. As the project is divided into various modules so the best approach to implement the project is to implement in a modular fashion.

TOP DOWN IMPLEMENTATION APPROACH:

A top-down approach is essentially meant to breaking down a system to gain insight into its compositional sub-systems. In a top-down approach an overview of the system is first formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in yet greater detail, sometimes in many additional subsystem levels, until the entire specification is reduced to base elements. A top-down model is often specified with the assistance of "black boxes" that make it easier to manipulate. However, black boxes may fail to elucidate elementary mechanisms or be detailed enough to realistically validate the model.

This is how the project is implemented first the administrator module is implemented and after that other modules are implemented as the data that has been added by the administrator module is being used in the other modules. If there is no data in the administrator interface then other modules will not work as the data in other module is based on the selection of the data from the administrator interface

.

INTERDEPENDENCY BETWEEN MODULES:

The modules of the application are interdependent as there is navigation from one module to another in the application. Also the data is being passed between modules. Initial data is to be input to the administrator module when first time the tool is implemented. After that other module can be implemented as the projects, practices, processes and their weight age is being generated from the administrator module. If there are no project initials in the lists then user module will not work. This states the interdependency of different modules on the administrator module. When the project is executed it will check for the users who are going to use the application.

There will be only one administrator of the tool who is from Ecologic Corporation only and he/she can access the whole application. Other users can access only their interface; if any person who is not administrator will try to access any type of data in the project he/she will not be able to see any type of editing option in the project as the administrator.

POST IMPLEMENTATION REVIEW:

After the project is implemented, a review should be conducted to determine whether the system is meeting expectations and where improvements are needed. Post implementation of this new proposed system will measure the system’s performance against predefined requirements. It determines how well the system continues to meet performance specifications. It also provides information to determine whether major or re-design or modification is required. Post implementation of proposed system is an evaluation of the system in term of the extent to which the system accomplishes the stated objectives and thus providing full consistency and reliability. Feedback and suggestions regarding changes and improvements are also asked from users.

**MAINTENANCE**

**MAINTENANCE**

The maintenance phase of software development involves debugging the application if there is a bug in the code. If the user makes changes in his/her requirements in future regarding the application, then making those changes also involved in software maintenance.

Till now all the requirements specified in the System Requirement Specification Document are fulfilled and the application is working properly as needed. Thus we can say that there is nothing likes a logical error or bug in the application. If in future there is any type of requirement change or to add any type of extra functionality or module to the project then the concerned industry can fulfill the requirements and user needs.

**PROJECT LEGACY**

**PROJECT LEGACY**

CURRENT STATUS:

Presently the project is working properly and it is ready to be implemented. This project meant a lot for me because I have enjoyed the great experience of the project exposure and problem solving. Hence, I can say, the basic functionality of the project is achieved which includes:

1 Very fast access to data

2. User-friendly environment

3. Ease in searching records or any information

4. Less error prone

5. Time saving

6. Graphical User Interface

REMAINING AREAS OF CONCERN:

There could be a future scope for the project of adding additional functionality or modules for other tasks being performed at a library. A provision could be made to run the system on a distributed system/networking environment. Additionally there could be a module for showing the report of particular student.

**REFERENCES AND BIBLIOGRAPHY**

**BIBLIOGRAPHY**

Books:

* VB.Net Black Book by Steven Holzner
* Dreamtech Evangelos Petroutsos VisualBasic.Net
* Teach Yourself VisualBasic.Net
* Database System Concepts by Silberschatz Korth Sudarshan (Tata McGraw Hills Publications)
* System Analysis and Designing by eias M.Award (Galgotia Publication)

FOR MORE DETAILS ABOUT THE PROJECT PLEASE CONTACT:

Divyam Dogra (BCA IIIrd G.P.G.C Bilaspur)

E-mail ID : divyamdogra@gmail.com