

Low Level Design (LLD)

College Library Management

Revision Number: 2.0

Last date of revision: 12/03/2023

Document Version Control

Date Issued	Version	Description	Author
06/03/2023	1	Initial LLD – V1.0	Ganesh k
12/03/2023	2	Updated KPI – V1.1	Ganesh k

Contents

Document Version Control	2
Abstract	4
1 Introduction	5
1.1 Why this Low-Level Design Document?	5
1.2 Scope	5
1.3 Definitions	5
2 General Description	6
2.1 Product Perspective	6
2.2 Problem statement	6
2.3 PROPOSED SOLUTION	6
2.4 FURTHER IMPROVEMENTS	6
2.5 Technical Requirements	6
2.6 Data Requirements	7
2.7 Tools used	8
2.7.1 Hardware Requirements	8
2.7.2 ROS(Robotic Operating System)	9
2.8 Constraints	9
2.9 Assumptions	9
3 Design Details	10
3.1 Process Flow	10
3.2 Reusability	12
3.3 Application Compatibility	12
3.4 Resource Utilization	12
3.5 Deployment	12
4 Conclusion	14

Abstract

A Library Management System is a system that is used to maintain the records of the library. It contains work like the number of the available books, the number of books issued, the number of books to return or renew. It helps to maintain a database that is useful to enter new books and records of books borrowed by the members with the respective submission dates. It will reduce the manual work done by the librarian to maintain the record of the library. It allows maintaining the resources in a more operative manner that will help to save the time. It is also convenient for the librarian to manage the process of books allocation. It is useful for students as well as a librarian to keep the constant track of the availability of all books in a library.

1 Introduction

1.1 Why this Low-Level Design Document?

The purpose of this Low-Level Design (LLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The LLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
 - Security
 - Reliability
 - Maintainability
 - Portability
 - Reusability
 - Application compatibility
 - Resource utilization
 - Serviceability

1.2 Scope

The LLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The LLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

<i>Term</i>	<i>Description</i>
<i>LMS</i>	College Library Management
<i>Database</i>	Collection of all the information monitored by this system(MYSQL)
<i>IDE</i>	Integrated Development Environment
<i>AWS</i>	Amazon Web Services

2 General Description

2.1 Product Perspective

The College Library Management is used by the collage students and collage management to issue the books from the collage library, it is fully automatic like no need for calculating fines day by day, the application it self has own feature to calculate the fine for excess dates. Also had two types of personal logins for those are admins and students as well.

Students only can see their profiles only , but the admin login can issue the book, delete book, add and delete the students and many..

2.2 Problem statement

Create a Web application to manage Library Information. Approach: Implement the below feature in your application.

- Student Registration.
- Book Management: Add a new book, delete a book, update a book's details, and search a book.
- Allow students a maximum of 15 days to borrow the book. If the book is not returned within 15 days, calculate the fine. One day equals ten rupees
- Limit each pupil to three books.
- Enable book searches by author, title, and category.
- Registration of librarians
- Any Librarian may issue books to any student.
- You can also create your own functions.

2.3 PROPOSED SOLUTION

This Online Library Management system will be an automated one. The users would be able to search up the book, check its availability. The admin department will be able to track the status of each user, including their details and books they have issued. So, this would increase the user's and admin's productivity by providing a better and simpler interface to manage the system. With the advent of technology, this system can be easily implemented by libraries especially in libraries of educational institutes.

2.4 Technical Requirements

Hardware Requirement

1. 4GB RAM
2. 1TB Hard Disk space in server machine
3. Core i5 or higher

Software Requirement

1. Windows 10 or above OS
2. SQL Server

2.5 Data Requirements

Signing In/Signup .:

Description (Scope) – The person has to sign in to his account, if he/she has already created an account.

The new users will have to sign up.

Input: **User ID and Password.**

Output: The requested page as per prior selections. Error message if wrong details are entered.

Input: Username, Password, e-mail, Contact Number.

Output: Creation of a new account, User ID and verification mail sent to the user.

Issuing of Books

Description (Scope) – The user searches for a particular book using the book name and author. The book is then searched in the database and the details of the book is displayed, including the availability and duration for borrowing. The user can then issue the book accordingly.

Search by name and author.

Input: Name and/or author of the book.

Output: Displays the book, author, publisher and availability status. Displays not found message if book is not present.

Input: Select the book and enter duration of borrowing.

Output: If user has any book overdue, the user is not granted permission to issue new books and is redirected to book returning page else redirected to next page.

Selecting a suitable time-slot.

Input: Select a suitable time-slot to collect the books to avoid crowding.

Output: The book is issued to the user, displaying a receipt with the respective details book code, User ID, due date for returning.

Returning of Books

Description (Scope) – The user selects which books to return and select a time-slot. If there is any fine, the user pays the fine accordingly.

Selecting the book.

Input: Choice of book(s) to return from their book list.

Output: Return prompt window.

Input: Selecting the return key.

Output: Respective page showing the details including book name, code, date of issue and overdue fine (if any).

Shows penalty prompt with amount if users have any book overdue.

The Low-Level Design Helps The Developers To Stay Focused Without Getting Deviated From The Deliverables. It Is One-Stop Reference Material For Them. The Document Is Divided Into Various Sections Based On The Different Factors Which Will Help In Developing A Robust Application That Is Of High Code Quality, Maintain Best Coding Practices, Incorporate Reusability, And Is Scalable.

Admin Management System

Description (Scope) – The admin department uses this feature for checking availability of particular books and borrowing history for each book. It also displays which user details with options to enable, disable. The admin department can also add new books and the respective details to the database.

Check availability status.

Input: Book code.

Output: Count of books available till date.

Input: Book code.

Output: The book borrowing history, displaying which particular copy has been borrowed by which user.

Updating the stock count of a book is shown in Home page in Fig 1.

Input: Book code, present stock count available or increment or decrement count.

Output: Count of that books gets updated.

Add new books to the database.

Input: Book details, including name, author, code, ISBN number and publisher, as well as the number of copies of each book.

Output: The page displaying the new list of books available.

Non-Functional Requirements

Correctness Requirement: This software performs accurately as intended and in no other way.

Portability requirement: This software provides a system-based interface to the user. Any device with Windows 7 or above can use it.

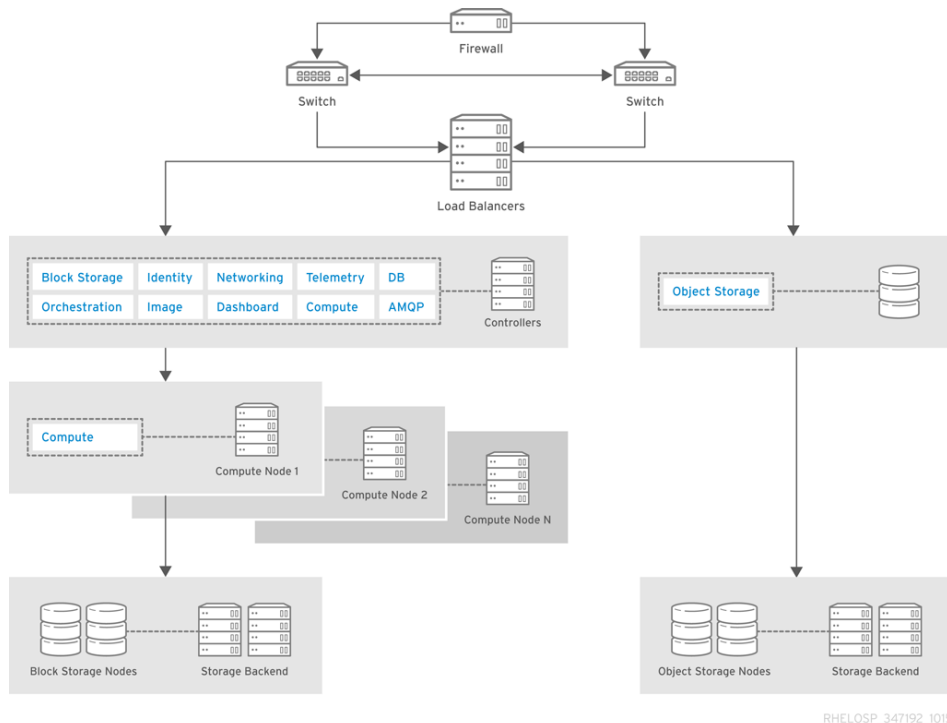
Efficiency Requirement: The software is highly efficient and various tasks in its various modules and sub-modules can be performed simultaneously.

Usability Requirement: The software has a simple but efficient user interface, which can be used by all types of users, both technically sound as well as people not having so much knowledge about technology. So, any user can use its functionalities without any sort of complications.

Availability Requirement: The software will be available at any time of the day. However, the physically collecting or returning of books can be done at working hours only.

Performance Requirement: The database can accommodate high number of articles and users without any fault. As the latest technologies have been used, so the system would be very responsive and the response would be extremely fast.

Reliability Requirement: The system is extremely reliable as there are proper measures to protect the data of the users, reviewers and the authors. Proper firewall and other security measures have been used to prevent any kind of breaching.



Low-Level Design Is A Detailed Description Of Every Module Of Software. It Describes Every Module In Detail By Incorporating The Logic Behind Every Component In The System. It Delves Deep Into Every Specification Of Every System, Providing A Micro-Level Design.

Low-Level Designs Are Created And Implemented By Designers And Developers. This Design Breaks Down High-Level Solutions Into The Smallest Details. In This Blog, We Will Elaborate On How To Create A Low Level Design.

The LLD Has Exhaustive Application Detailing That Concentrates On How The Different Parts Of The Unit Will Work Together. It Gets Into The Details Of How The Components And Classes Work, The Various Properties Of The Classes, The Definitions Of The Database, And The Interfaces.

Often The Developers Rush Through The Low-Level Design Phase To Save Time Only To End Up Spending That Time On Resolving The Errors Due To The Shallow Design Process. Here, At WalkingTree The BRD (Business Requirement Document) Acts As The High-Level Design Document Which Has The Architectural Diagrams, Data Flow Diagrams, And Sequence Diagrams Included.

When We Approach Low-Level Design, The Idea Is To Get Into Further Detailing Of The Application With Reference To The BRD. The Low-Level Design Document Is Often A Heavy-Duty, Bulky Document That Contains The Pseudocode For The Developers. The Main Objective Of Preparing Low-Level Design Is To Mitigate Any Additional Burden Developers Face During The Development Phase By Clearly Laying Out Details. The LLD Is Essentially A Go-To Document For The Developers; Reference Material With All The Technicalities Involved, A Well Laid Out Map, With The Intention To Make Development A Very Smooth Journey.

2.6 Tools used

Java programming language and JDBC, J2EE Servlets, JSP, JS, JSTL ,HTML, CSS and MYSQL database are used to build the whole model.



- Eclipse is used as IDE.
- Tomcat is used for deployment of the model.
- MySQL is used to retrieve, insert, delete, and update the database.
- Front end development is done using HTML/CSS/JS
- J2ee, java, servelets is used for backend development.

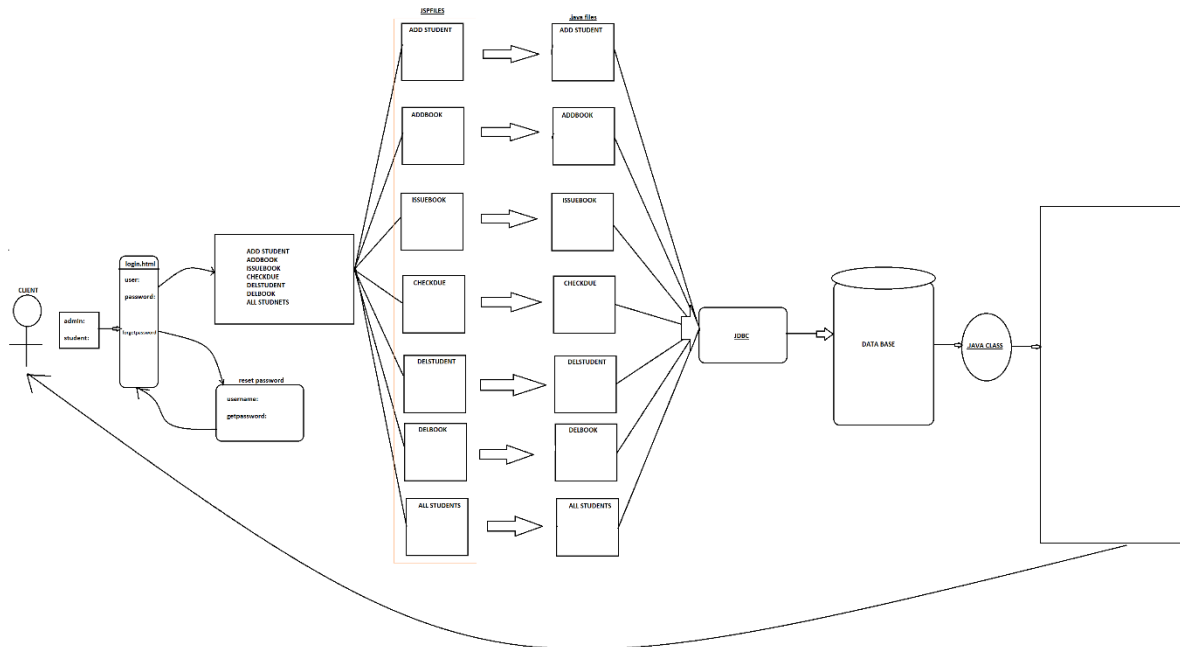
2.7 Assumptions

- Coding is error free.
- System has specified hardware and software requirements.
- Fast access to database.

- User does not provide any incorrect information.

3 Design Details

3.1.1 Model Training and Evaluation



3.1 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.2 Reusability

The code written and the components used should have the ability to be reused with no problems.

3.3 Application Compatibility

The different components for this project will be using Java as an interface between them. Each component will have its own task to perform, and it is the job of the java to ensure proper transfer of information.

3.4 Deployment

Tomcat used to deployment for localhost.

4. Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong?

An error will be defined as anything that falls outside the normal and intended usage.

Conclusion

With the evolution of technology and it being so blended in our daily lives, it is imperative that we discard time-consuming laborious methods to implement something which would be so clean and compact to use through computers. This system provides efficient service to the various users. Implemented with the best technology available, this software is convenient to use and virtually fault-free, providing the users with a smooth and unique experience.

