Low Level Design

Online Book Shop

|  |  |
| --- | --- |
| Written By | Akash Kshirsagar |
| Document Version | 0.1 |
| Last Revised Date | 25 – feb -2024 |

**Document Control**

**Change Record:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 0.1 | 25-feb-2024 | Akash Kshirsagar | Document Content , Version Control and Unit Test Cases to be added |

**Reviews:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reviewer** | **Comments** |
| 0.1 | 3 – march -  2024 | Akash Kshirsagar | Document Content , Version Control and Unit Test Cases to be added |

**Approval Status:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Review Date** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |

Contents

[**1.** **Introduction** 1](#_Toc15828)

[**1.1.** **What is Low-Level design document?** 1](#_Toc15829)

[**1.2.** **Scope** 1](#_Toc15830)

[**2.** **Architecture** 2](#_Toc15831)

[**3.** **Architecture Description** 3](#_Toc15832)

[**3.1.** **Presentation Layer** 3](#_Toc15833)

[**3.2.** **Business** **Logic** **Layer** 3](#_Toc15834)

[**3.3. Data Access Layer** 3](#_Toc15835)

[**4.** **Requirements Gathering** 4](#_Toc15836)

**4.1. System Design** 4

**4.2. Implementation** 4

**4.3. Testing** 4

**4.4. Deployment** 4

# 1. Introduction

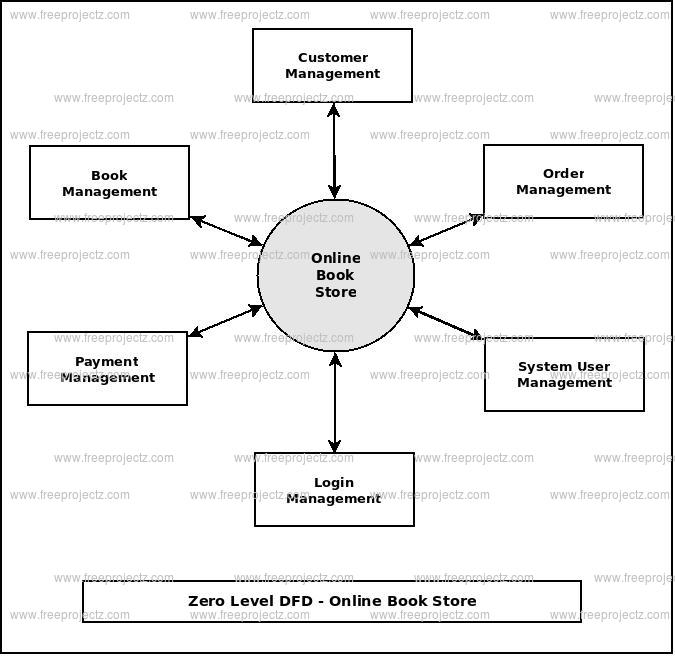
## 1.1.What is Low-Level design document?

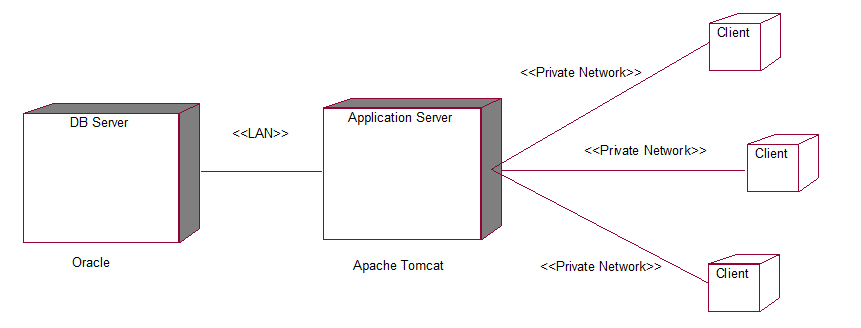
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

## 1.2.Scope

Low-level design (LLD) is a component-level design process that follows a step-bystep [refinement](https://en.wikipedia.org/wiki/Refinement_(computing)) process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

# 2.Architecture





# 

# 3. Architecture Description

## 3.1. Presentation Layer:

Utilizes JSP for dynamic content generation.

HTML/CSS for frontend layout and styling.

Bootstrap for responsive design.

## 3.2. Business Logic Layer:

Servlets handle HTTP requests and responses.

Implements business logic for book management, order processing, etc.

## 3.3. Data Access Layer:

Utilizes JDBC (Java Database Connectivity) for interaction with MySQL database.

Manages database operations such as CRUD (Create, Read, Update, Delete) for books, users, orders, etc.

# 4. Requirements Gathering:

### Gather user requirements and define system functionalities.

## 4.1. System Design:

Design system architecture, database schema, and user interface wireframes.

# 4.2. Implementation:

Develop backend logic using Java, Servlets, and MySQL for database interactions. Implement frontend using JSP, HTML/CSS, and Bootstrap.

# 4.3. Testing:

Conduct unit testing, integration testing, and user acceptance testing to ensure system reliability and functionality.

## 4.4. Deployment:

Deploy the application on a web server such as Apache Tomcat for production use.