

# HLD and LLD

In System Design

## Introduction

*When embarking on a software development journey, architects and developers follow a systematic approach to transform ideas into a fully functional and efficient application.*

*Among the crucial steps in this process are High-Level Design (HLD) and Low-Level Design (LLD).*

*These two distinct phases play a pivotal role in bridging the gap between the conceptualization of a software system and its actual implementation.*

## High-Level Design (HLD)

At the outset of any software development project, architects engage in the High-Level Design phase. HLD serves as the blueprint of the application and provides an abstract view of the system. The primary objectives of HLD are as follows:

- 1. Architectural Overview:** During HLD, the software's overall architecture is conceptualized, defining its major components, modules, and their interconnections. This high-level representation helps stakeholders grasp the system's structure without delving into intricate technical details.
- 2. Functional Specifications:** HLD outlines the functionality and behavior of each component, emphasizing what each part should do rather than how it should be accomplished. This ensures that the software fulfills its intended purpose effectively.
- 3. System Interactions:** The interaction between different modules and external systems is explored in HLD. This includes defining input and output interfaces and specifying how various components will communicate with each other.

**4. Performance Considerations:** Architects also consider the software's performance requirements in the High-Level Design phase. Factors like response times, processing speed, and scalability are evaluated to lay the groundwork for an efficient system.

**5. Technology Stack Selection:** HLD aids in determining the appropriate technology stack and tools that will be used to implement the system effectively.

### **Low-Level Design (LLD)**

After the completion of High-Level Design, developers move on to the Low-Level Design phase. LLD focuses on the nitty-gritty details of the system, drilling down from the abstract representation of HLD to a more concrete and technical level. The main objectives of LLD include:

**1. Detailed Component Specifications:** LLD delves into the detailed specifications of each module and component. It involves defining data structures, algorithms, and the specific operations each module will perform.

**2. Interface Definitions:** During LLD, the interfaces between different components are precisely defined. This ensures that each module communicates efficiently with others, promoting modularity and maintainability.

**3. Resource Allocation:** LLD addresses resource allocation and management, such as memory usage, CPU utilization, and database storage.

**4. Error Handling and Recovery:** In this phase, architects also plan for error handling and recovery mechanisms to ensure that the system remains robust and reliable even in the face of unforeseen errors.

**5. Security Considerations:** LLD incorporates security measures, such as access controls, encryption, and data validation, to safeguard the system against potential threats.

**6. Code Structure:** Developers structure the actual codebase during LLD, aligning it with the defined architecture and design patterns.

High-Level Design	Low-Level Design
Abbreviated as HLD	Abbreviated as LLD
Understanding the flow between different system constituents requires HLD. For instance, HLD explains how everything functions as a single organism if you have numerous connected solutions.	LLD describes how to create the product, configure it, and troubleshoot it. For instance, if you have multiple solutions, LLD goes into great detail on each one, just like an organ in the body.
HLD is also known as Macro Level Design.	LLD is also known as Micro Level Design.
HLD converts Business Requirements into High-Level Solutions.	LLD converts High-Level solutions to Detailed Solutions.
HLD contains system and solution architecture, use cases, data flow, and interfaces.	LLD contains feature design, impact analysis, low-level design, pseudo code, <b>algorithms</b> , interface specifications, and <b>design considerations</b> .
Software projects' HLD is often created in the early phases, usually during the SDLC's Analysis phase.	Required after HLD to have a detailed description of every module.

HLD is created by solution architects.	Designers and developers create LLD.
Clients, review teams, and design teams are the participants in a high-level design.	The operation teams, design teams, and implementers are involved in the low-level design.
SRS, i.e., Software Requirement Specification, is the input criteria in HLD.	Reviewed HLD is the input criteria for LLD.
The business or client requirement is transformed into a high-level solution via high-level solutions.	The High-Level Solution is transformed into a Detailed Solution by Low-Level Design.
High-Level Design results in review record, function design, and database design.	Low-Level Design results in program specification and unit test plan