

LOW LEVEL

DESIGN

NAME: SAHIL PARAG KINDO

What is Low-Level design document?

Detailed Technical Information: LLD documents delve into the technical specifics of individual components, subsystems, or modules within a software system. They describe how these components will be built, how they interact, and how they will be implemented in code.

Granular Design Details: LLD documents provide a granular view of the software's internals, including data structures, algorithms, database schema, interfaces, and functions or methods.

Design Rationale: LLD documents often include explanations for design choices. This helps developers and other stakeholders understand why certain decisions were made and how they contribute to achieving project goals.

Coding Guidelines: LLD documents may specify coding conventions, standards, and best practices that developers should follow when implementing the design. This ensures consistency and maintainability of the codebase.

Development Roadmap: LLD documents can outline the step-by-step process for implementing the component, including dependencies on other components, development priorities, and milestones.

- Introduction

Purpose of the Document

Overview of the Chat Application

- Client-Side Components

User Interface (UI) Design

Chat List Interface

Chat Window Interface

Message Input Interface

Client Logic

User Authentication

Message Handling

User Interface Interaction

Communication Layer

WebSocket Integration

Message Encoding and Decoding

Error Handling

- **Server-Side Components**

Application Server

Request Handling

User Session Management

Database Interaction

Database

Schema Design

User Data Management

Message History Storage

Message Broker (if used)

Message Distribution Logic

Queue Management

- **Security Measures**

Authentication and Authorization

User Authentication Process

User Authorization and Access Control

Data Security

Data Encryption (e.g., HTTPS)

Input Validation

Handling Security Threats

- Real-Time Messaging

WebSocket Integration

Client-Side WebSocket Implementation

Server-Side WebSocket Handling

Message Queuing (if used)

Message Routing

Scalability and Load Balancing

- User Profiles

User Profile Management

User Registration

Profile Updates

Profile Picture Handling

- Group Chats

Group Chat Creation and Management

Creating Groups

Inviting Users to Groups

Managing Group Memberships.

- **Message History**

Retrieving Message History

User-Specific History

Group Chat History

Pagination and Searching

- **Notifications**

Notification Handling

New Message Notifications

Online Status Notifications

- **Error Handling**

Error Codes and Messages

Logging and Debugging

Exception Handling Strategies

- Testing and Quality Assurance

Unit Testing

Integration Testing

User Acceptance Testing

Test Cases and Scenarios

- Deployment

Infrastructure Setup

Deployment Strategy

Server Configuration

Monitoring and Scaling Plan

- Documentation

Code Documentation

API Documentation

User Manuals (if applicable)

- Conclusion

Summary of Key Design Points: Begin by summarizing the most crucial design decisions and technical details covered in the document. This should be a concise overview to remind readers of the essential aspects of the design.

Design Rationale: Mention why specific design choices were made and how they align with the project's goals and requirements. Emphasize the value and benefits of these choices.