



Data Base Server

High Level Design

Document Control:

Project Revision History

Date	Version	Author	Brief Description of Changes

Team Members

Employee ID	Name
46290158	Muskan Pathan Allabaksh
46282451	Sri Harshita Talari
46282452	Renuka Gujjala
46282098	Bhagyalakshmi Daliparthi
46282102	Sarika Pathiputturu
46290159	Vinitha Devalla

Table of Contents

1.Introduction	
1.1	Purpose
1.2	Scope
1.3	Overview
2. General Description	
2.1	Product Perspective
2.2	Tools Used
2.3	General Constraints
2.4	Assumptions
2.5	Special Design aspects
3.Design Details	
3.1	Main Design Features
3.2	Data Flow Diagram (Level-0)
3.3	Data Flow Diagram (Level-1)
3.4	User Interface
3.5	Error Handling
3.6	Help
3.7	Performance
3.8	Reliability
3.9	Maintainability
3.10	Portability
3.11	Reusability
3.12	Application Compatibility

High Level Design

1.Introduction

1.1 Purpose

The purpose of this High Level Design (HLD) 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.

1.2 Scope

This HLD document provides the detailed overview of the Database Server. It highlights the high level flow of transmission of data between clients and server as input to the low level design document that would further elaborate on the proposed system design.

1.3 Overview

This HLD Document is arranged in the following format:

Section1: Introduction

A brief explanation about purpose, aim, scope, and design format of the proposed project.

Section 2: General Description

This section is all about the general constraints, assumptions, and design aspects associated with the proposed project. The product perspective will give an overall description of the Database Server.

2. General Description

2.1 Product Perspective

The Database Server is based on the concept of communication between client and server and managing the data in database. The server will maintain the different type and formats of data in database and will receive the relative data from client and will add to the database. The server can connect to single client at a time. No two clients can access or modify the same database at a time. The server should employ proper connection of database in case two clients attempt to modify the database values at same time.

2.2 Tools Used

1. Valgrind
2. Make

2.3 General Constraints

1. User will give the input data to client.
2. We are considering only two clients.

2.4 Assumptions

The main goal of this project is to maintain the protection of data which is present in the database when two clients are trying to modify it. In doing so, many documents are created, and it is assumed that design flaws will be found early on. It is also assumed that all aspects of this project have the ability to work together.

2.5 Special Design aspects

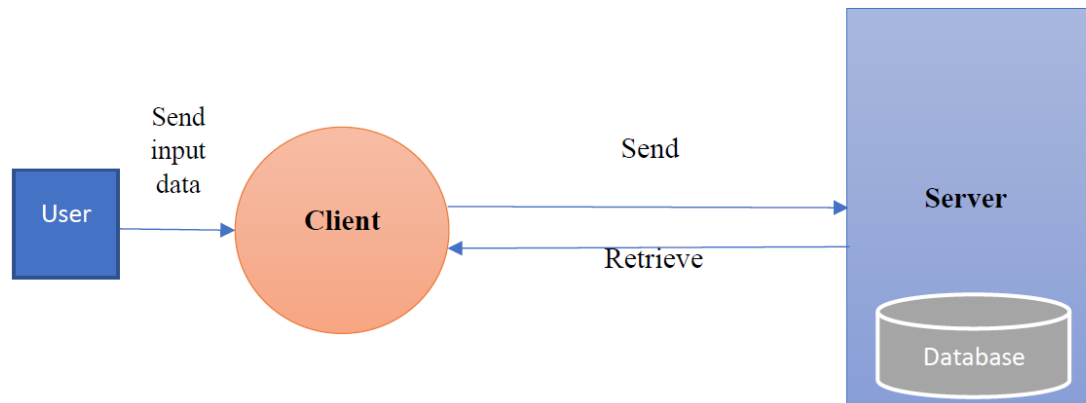
One of the design aspects is that the server will use threads to maintain different types and formats of data.

3. Design Details

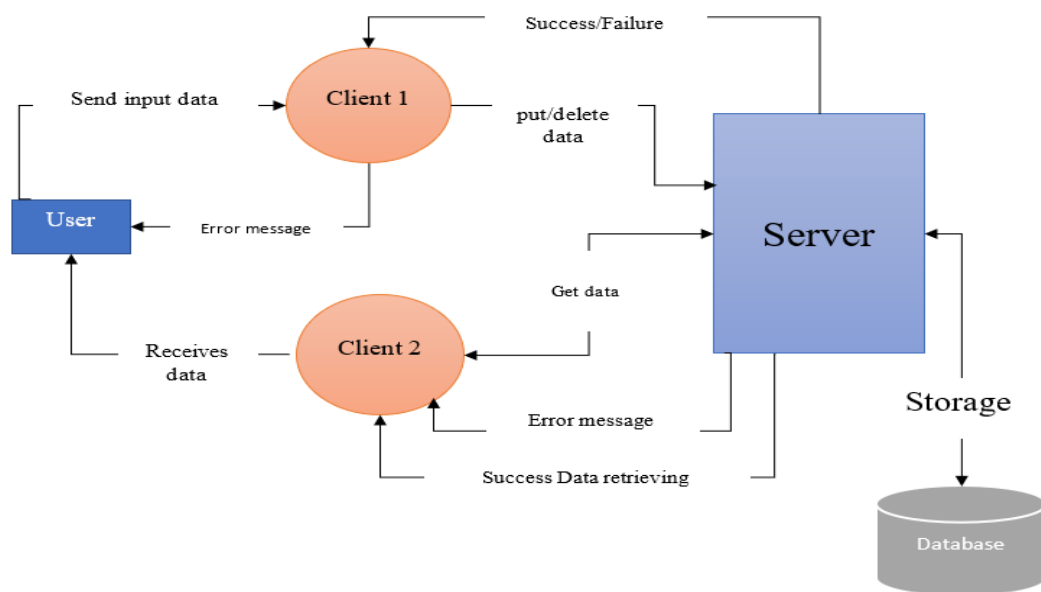
3.1 Main Design Features

The main design features include following major parts: the clients and server, the database, and the user. In order to make these designs easier to understand, the design has been illustrated in attached diagram (Data Flow Diagram)

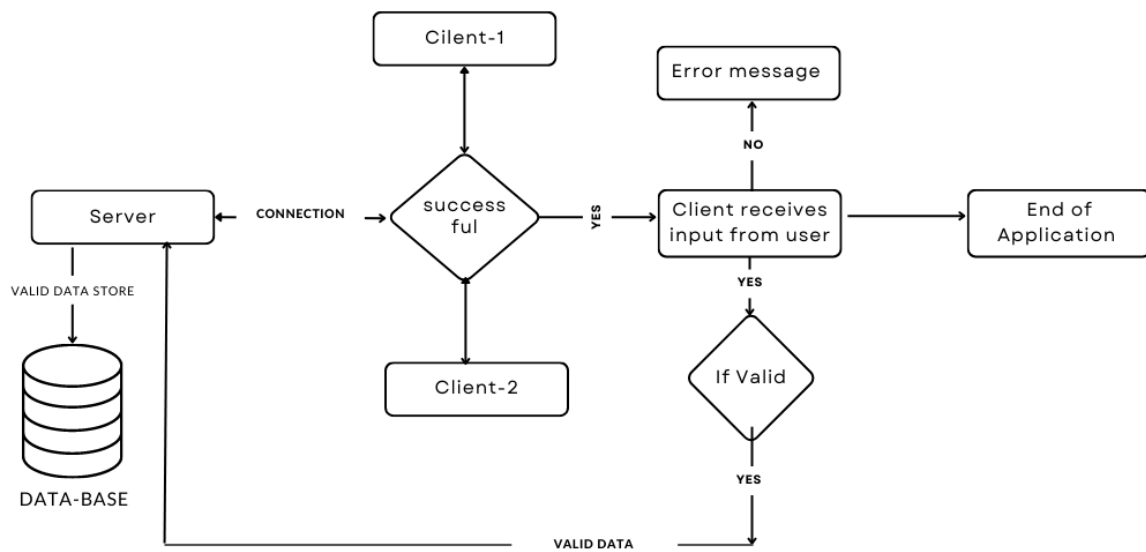
3.2 Data Flow Diagram (Level-0)



3.3 Data Flow Diagram (Level-1)



3.4 High Level Design



3.5 User Interface

Menu Driven Interface

3.6 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.

3.7 Help

Help will come in the form of all the documentation created prior to coding, which explains the intended uses. Should time allow, detailed instructions will be written on how to create and implement the system with the intention of publishing as an Open Source solution.

3.8 Performance

Performance is going to be very important for this project. For everything to run smoothly for this project, The server maintain the restrictions to the type and format of data given by the user and should maintain the consistency of data in the database.

3.9 Reliability

The server can protect the database without being duplicated and maintain different types of data.

3.10 Maintainability

NA

3.11 Portability

This system should have the ability that, once it is together, the entire system should be able to be physically moved to any location. Code and program portability should be possible between kernel-recompiled Linux distributions. For everything to work properly, all programs should be in one folder.

3.12 Reusability

The code has the ability to be reused with no problems. Everything will be completely reusable to anyone.

3.13 Application compatibility

This was designed as an independent system. As it is not connected to any other components or interfaces, application compatibility is not a concern.