Capgemini

HOTEL RESERVATION/MANAGEMENT SOFTWARE

High Level Design Version Draft v0.4

Document Control :

Project Revision History

Date	Version	Author	Brief Description of changes

Team members

Employee id	Name
46282418	Arveti Chandana Gayathri
46282420	Sian Gijo
46282421	Manisha L
46290138	Kappa Latha
46290139	Karishma Shaik
46290156	Anusha Nallam

Table of Contents

High Level Design

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 document provides a comprehensive high level design overview of the Hotel Management/ Reservation system. It highlights the high level flow of the functionalities of the system and serves as an input to the low level design documents 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 the 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 reservation system.

- Section 3 : Design Details

This section documents the detailed design of all modules associated with the development of the proposed hotel reservation system.

2. General Description

2.1 Product Perspective

The project, Hotel Management System is a client server based application that allows the server/manager to handle all hotel activities. Users/Clients are required to authorize themselves with username and password to access hotel data from the server. Client requests booking of a certain room type such as single, double, deluxe or suite room along with the check in and check out date. Server checks for the availability of the same and initiates the booking once confirmed by the client. Server also maintains booking history up to a month and displays statistics related to room availability.

2.2 Tools used

1. C and System programming tools.
2. Lucid chart application is used for pictorial representations.

2.3 General Constraints

1. The user needs to be authorized by the server before logging into the system.
2. We are using mutex to provide synchronization in shared memory.

2.4 Assumptions

We are assuming that the user will display the hotel rooms and services in a menu driven format.

2.5 Special Design aspects

One of the design aspects is that the system will work with two clients at a time.

3. Design Details

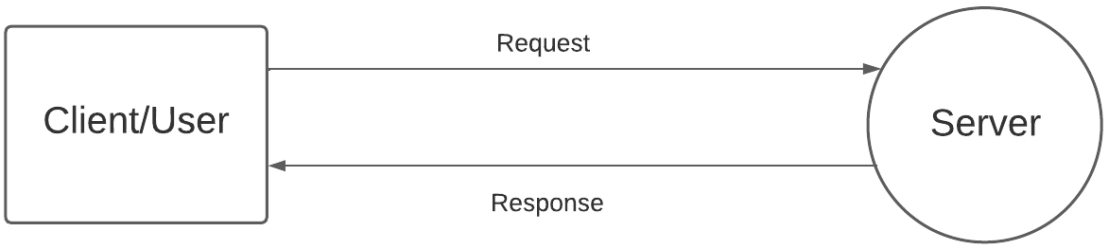
3.1 Main Design Features

The main design features include four major parts: the architecture, the user interface design, the files and functionalities. In order to make these designs easier to understand, the design has been illustrated in attached diagrams.

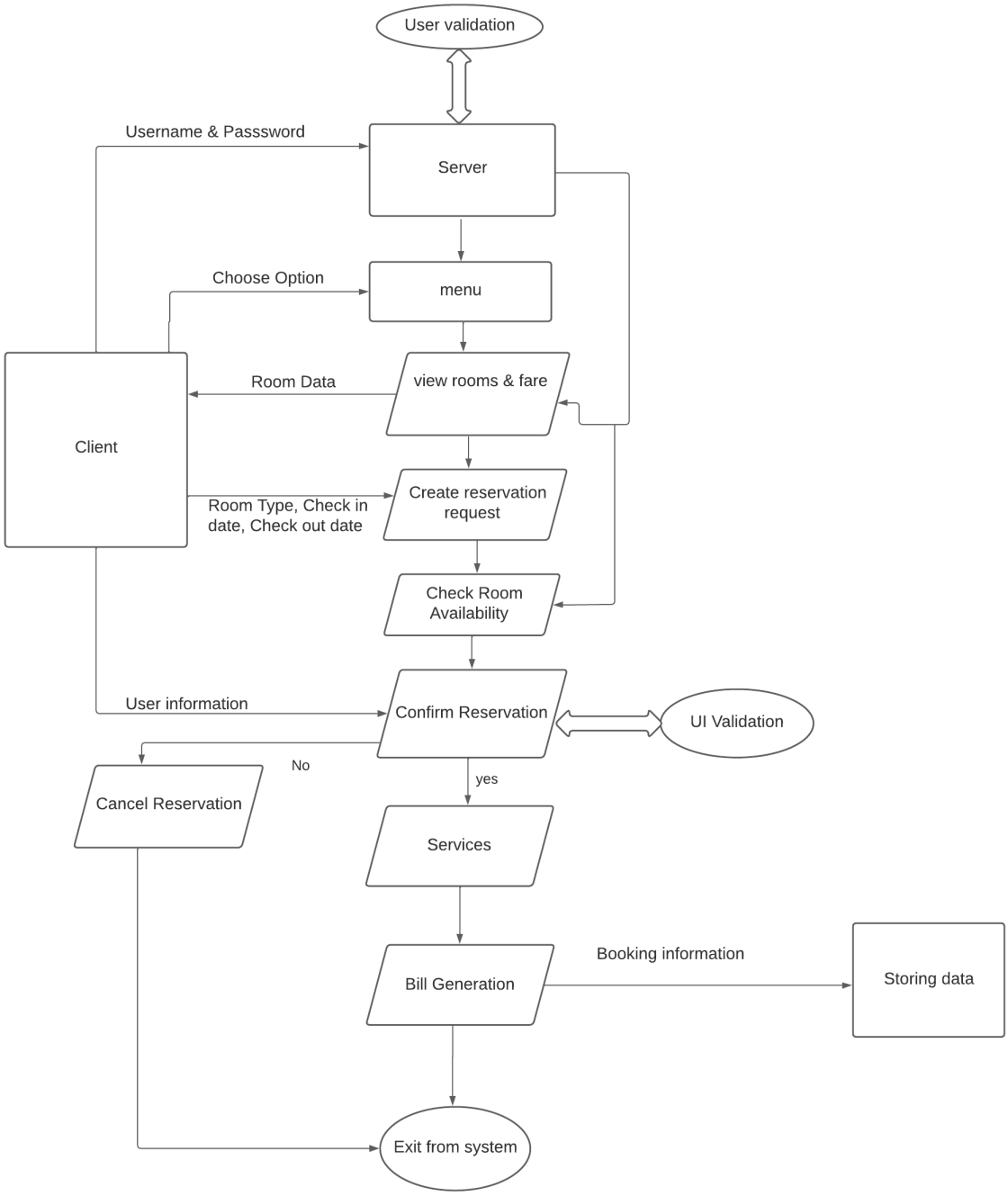
3.2 Standards

- Security –NA
- Quality – by keeping the interface simple and direct, quality should be kept at a maximum**.**

3.3 Data Flow Diagram (Level - 0)

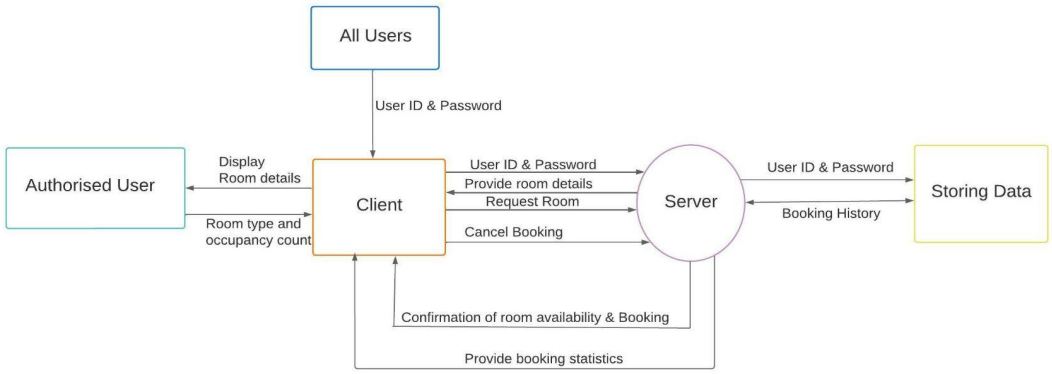


3.4 Data Flow Diagram (Level - 1)



Description:

3.5 High Level Design



3.6 User Interface

Menu Driven interface.

3.7 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.

3.8 Help

Help will come in the form of all the documentation created prior to coding, which explains the intended uses. Source code can be created by including proper comments that describe the functionality being implemented.

3.9 Performance

Performance is going to be very important for this project. For everything to run smoothly for this project, The Hotel Management System will work on the user terminal, performance depends upon the software of the user and The reservation is done by the user.

3.10 Reliability

The system can handle multiple clients at a time and ensure efficient allotment and booking of rooms without creating errors.

3.11 Maintainability

NA

3.12 Portability

Code and program portability should be possible between kernel-recompiled Linux distributions. For everything to work properly, all programs should be in one folder.

3.13 Reusability

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

3.14 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.