Capgemini

HOTEL RESERVATION/MANAGEMENT SOFTWARE

Low-Level Design Version Draft v0.2

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

I) Low-Level Design

Low-Level Design

1. Introduction

The aim of this document is to gather, analyze and give an in-depth insight into the complete Hotel Reservation/Management software by defining the problem statement in detail.The intended audience include developers,project managers, hotel managers etc. The detailed low-level design of the Hotel Reservation System is provided in this document.

1.1 Purpose

The purpose of this document is to describe the low-level design flow of the Hotel Management/ Reservation software.

1.2 Document Conventions

TBD (To be continued).

1.3 Intended Audience and Reading Suggestions

The document is primarily intended for team members, which consists of trainees under the Capgemini Training program.

The references are:

1. System Requirements Specification Document

2. Detailed System Design

2.1 Design Descriptions:

This project design mainly focuses on implementing the Hotel Management/ Reservation system. Users will be able to choose services given by the hotel. Reservation of rooms of a particular type can be made by requesting the server that checks its availability and initiates the booking.

User Authentication menu:

The user will be displayed a menu to login or sign up. If the user is new to the system , he/she should sign up by setting up the username and password. The user authentication details will be stored in a file. If the user is an existing user, he/she should login by entering the username and password which will be authenticated and the next main menu will be displayed.

Main menu:

- The main menu has options to view rooms, book rooms, cancel rooms and quit the system.
- Depending on the choice entered by the user, corresponding functions are performed
- For choice 1, the server will display the type of rooms, price and services provided by the hotel to the user.
- For choice 2, the client will provide the room type, check in and check out date to book a room, The server receives this information and checks if the room of that particular type is available, If available the server takes confirmation from the user and books the room. The user will also be able to add-on the services provided by the hotel. The server then generates the bill and displays it to the user. The server will also store the user details in a file which maintains the booking history.
- For choice 3 , the user will be able to cancel the booked room, The client sends the room number and username to the server, the server will deallocate the room for that user and send a message to the client.
- For choice 4 , the client quits from the system.

2.3 Modules

2.3.1 client .c

main() :

Name	main			
Input	Parameter Name :	NA	Initial value:	NA
Output	Return value type:	NA	-	-
Description	The main function will display user to login or sign up to the system			
Pseudo Code	1. Display the main menu 2. Wait for user options 3. Based on user input it will move to another			

2.3.2 signup () :

Name	signup()			
Input	Parameter Name:	char name[] char username[] char password[]	Initial Value:	NA
Output	Return value type :	void	-	-
Description	This program is used to sign in the user to the system. It takes in the user's name, username and password from the user and saves it into a file.			

2.3.2 login() :

Name	login.c()			
Input	Parameter Name:	char name[] char username[] char password[]	Initial Value:	NA
Output	Return value type :	int	-	-
Description	This program is used to login in the user to the system, It takes in the user’s name, username and password from the user and checks whether the username and password is valid , If valid it returns 1 else returns 0			

2.3.3 main_menu() :

Name	main_menu()			
Input	Parameter Name	NA	Initial value:	NA
Output	Return value type	NA	-	-
Description	This list of options to view rooms and services, book room, cancel booking and quit will be displayed to the user			
Pseudo Code	1. Display the main menu 2. Wait for user options			

2.3.4 display rooms()

Name	Display rooms_services()			
Input	Parameter Name	NA	Initial value:NA	-
Output	Return value type	NA	-	-
Description	Displays the room types , their price for 24 hours and the services provided by the hotel			
Pseudo Code	1.Take the information from the server 2. Display the room types, price and the services to the user			

2.3.4 display services()

Name	Display services()			
Input	Parameter Name	NA	Initial value:NA	-
Output	Return value type	NA	-	-
Description	Displays the services provided by the hotel			
Pseudo Code	1. Display the services provided by the hotel			

2.3.5 book_rooms()

Name	**book_rooms()**			
Input	Parameter Name	NA	Initial value:NA	-
Output	Return value type	NA		-
Description	Prompts the user to provide the room type, check in and check out date and initialize it to structure and send the information to the search that particular room type funtion			
Pseudo Code	1. Prompts the user to enter room type , check in and check out date 2. If room_type is single it navigates to reserve_single_room() function 3. If room_type is double it navigates to reserve_double_room() function 4. If room_type is delux it navigates to reserve_delux_room() function 5. If room_type is suite it navigates to reserve_suite_room() function			

2.3.6 reserve_single_room()

Name	**reserve_single_room()**			
Input	Parameter Name	char *user	Initial value:NA	-
Output	Return value type	Boolean		-
Description	Searches the room in hotel structure, if not occupied , set to occupied , update the check in and check out and set the username inside the room structure and returns 1 if success or returns 0			
Pseudo Code	1. Take mutex 2. Search the single_room array in hotel structure 3. If bool occupied is 0 , set to it to 1 and update check in ,check out date 4. If bool occupied is 1 , return 0 5. Set username in the room structure 6. Unlock mutex 7. Return 1			

2.3.6 reserve_double_room()

Name	**reserve_double_room()**			
Input	Parameter Name	char *user	Initial value:NA	-
Output	Return value type	Boolean		-
Description	Searches the room in hotel structure, if not occupied , set to occupied , update the check in and check out and set the username inside the room structure and returns 1 if success or returns 0			
Pseudo Code	8. Take mutex 9. Search the double_room array in hotel structure 10. If bool occupied is 0 , set to it to 1 and update check in ,check out date 11. If bool occupied is 1 , return 0 12. Set username in the room structure 13. Unlock mutex 14. Return 1			

2.3.6 reserve_delux_room()

Name	**reserve_single_room()**			
Input	Parameter Name	char *user	Initial value:NA	-
Output	Return value type	Boolean		-
Description	Searches the room in hotel structure, if not occupied , set to occupied , update the check in and check out and set the username inside the room structure and returns 1 if success or returns 0			
Pseudo Code	15. Take mutex 16. Search the delux_room array in hotel structure 17. If bool occupied is 0 , set to it to 1 and update check in ,check out date 18. If bool occupied is 1 , return 0 19. Set username in the room structure 20. Unlock mutex 21. Return 1			

2.3.6 reserve_suite_room()

Name	**reserve_suite_room()**			
Input	Parameter Name	char *user	Initial value:NA	-
Output	Return value type	Boolean		-
Description	Searches the room in hotel structure, if not occupied , set to occupied , update the check in and check out and set the username inside the room structure and returns 1 if success or returns 0			
Pseudo Code	22. Take mutex 23. Search the suite_room array in hotel structure 24. If bool occupied is 0 , set to it to 1 and update check in ,check out date 25. If bool occupied is 1 , return 0 26. Set username in the room structure 27. Unlock mutex 28. Return 1			

2.3.7 ()

Name				
Input	Parameter Name	NA	Initial value:NA	-
Output	Return value type	int		-
Description				
Pseudo Code				

2.3.8 ()

Name				
Input				-
Output				-
Description				

2.3.9 ()

Name				
Input				-
Output			-	-
Description				
Pseudo Code				

2.3.10 ()

Name				
Input				
Output				
Description				
Pseudo Code				

2.5 Structure used

- **Proc structure was created to store parameters for each entered binary.**

```
typedef struct
{ char name[MAX_SIZE];
//For storing name of entered binary
        char path[MAX_SIZE];
//For storing path of the entered binary
        int time_req;

//For storing burst_time of entered binary (if applicable)
        int prio;


//For storing priority of entered binary (if applicable)
} proc;
```

2.3 Modules

2.3.1 server.c

main()

Name				
Input	Parameter Name		Initial value	-
Output	Return value type		-	-
Description				
Pseudo Code				

2.3.2 ()

Name				
Input	Parameter Name			
Output	Return value type			
Description				
Pseudo Code				

2.3.3 ()

Name				
Input	Parameter Name		Initial value:	-
Output	Return value type			-
Description				
Pseudo Code				

2.3.4 ()

Name				
Input	Parameter Name	NA	Initial value:NA	-
Output	Return value type	int		-
Description				
Pseudo Code				

2.3.5 ()

Name				
Input	Parameter Name	NA	Initial value:NA	-
Output	Return value type	int		-
Description				
Pseudo Code				

2.3.6 ()

Name				
Input	Parameter Name	NA	Initial value:NA	-
Output	Return value type			-
Description				
Pseudo Code				

2.3.7 ()

Name				
Input	Parameter Name	NA	Initial value:NA	-
Output	Return value type	int		
Description				
Pseudo Code				

2.3.8 ()

Name				
Input			Initial value:NA	
			Initial value:NA	
			Initial value:NA	
Output				
Description				

2.3.9 ()

Name				
Input			Initial value:NA	-
			Initial value:NA	
Output				
Description				

2.3.10 only_name()

Name				
Input				
Output				
Description				
Pseudo Code				

2.5 Structure used

- **Proc structure was created to store parameters for each entered binary.**

// Todo//