

Hotel Reservation System Documentation

1. Project Overview

The Hotel Reservation System is a console-based application designed to manage hotel bookings, user accounts, and room details. The system supports both Admin and User/Customer roles with different access levels:

- Admin: Full control over rooms, bookings, reports, and user accounts.
- User/Customer: Can create accounts, search for rooms, book rooms, and cancel bookings.

The project uses text files for persistent storage:

- Accounts.txt → User account details
- All Rooms.txt → Room details and status
- Booked Rooms.txt → Booking information

The system is built in C++ using OOP, file handling, and inheritance.

2. Features

User/Customer Features

1. Create an account
2. Log in
3. Search available rooms
4. Search rooms by type (Single, Double, Suite)
5. Book a room
6. Cancel a booking

Admin Features

1. Admin login
2. View all booked rooms
3. Add a new room
4. Delete a room
5. Generate hotel reports:
 - Total rooms, booked rooms, available rooms

- Most and least booked room types
 - Total revenue and occupancy rate
 - Registered users
6. View all registered accounts
7. Delete a user account

3. Classes and Functions

3.1 User Module

Class: User

- Attributes: name, username, password, email
- Functions:
 - `newaccount()` → Create a new user account and save to Accounts.txt
 - `login()` → Verify username and password from Accounts.txt

Class: SearchAvailable (inherits User)

- Function: `search()` → Display rooms from All Rooms.txt where status is "Available"

Class: BookRoom (inherits User)

- Functions:
 - `searchbytype()` → Display rooms by type from Single.txt, Double.txt, suite.txt
 - `bookdetail()` → Book a room:
 1. Check room existence and availability
 2. Collect user info: name, contact, CNIC, guests, dates, nights
 3. Calculate total cost
 4. Update Booked Rooms.txt
 5. Change room status in All Rooms.txt from "Available" → "Booked"

Class: cancell (inherits User)

- Function: `Cancell()` → Cancel a booking:
 1. Remove booking from Booked Rooms.txt
 2. Update room status in All Rooms.txt from "Booked" → "Available"

3.2 Admin Module

Class: adminlogin

- Function: Adminlogin() → Verify admin credentials (username: admin, password: 931357)

Class: Admin

- Function: viewbooking() → Display all booked room details from Booked Rooms.txt

Class: addroom (inherits Admin)

- Function: AddRoom() → Add a new room to All Rooms.txt

Class: DeleteRoom (inherits Admin)

- Function: DELroom() → Delete a room from All Rooms.txt by room number

3.3 Reporting & Account Management

Class: Reports (inherits Admin)

- Function: reports() → Generates detailed hotel reports:
 - Total, booked, and available rooms
 - Single, Double, Suite bookings
 - Most and least booked room types
 - Total revenue
 - Occupancy rate
 - Total registered users

Class: AccountDetails (inherits Admin)

- Function: account() → Display all registered accounts

Class: deleteacc (inherits Admin)

- Function: Delete() → Delete a user account by name from Accounts.txt

4. File Structure

File Name	Purpose
Accounts.txt	Stores user accounts (Name, Username, Password, Email)
All Rooms.txt	Stores all rooms with details and status
Booked Rooms.txt	Stores booking information including guest details and total cost
Single.txt	Stores details of Single rooms
Double.txt	Stores details of Double rooms
suite.txt	Stores details of Suite rooms

5. Main Logic Flow

5.1 Program Start

1. User selects role: Admin or User
2. Admin → Login → Access admin menu
User → Create account or login → Access user menu

5.2 Admin Menu Flow

[Admin Login Successful]

-
- | | |
|-----------------|--|
| 1.Search Rooms | |
| 2.Book Room | |
| 3.View Bookings | |

- | 4.Cancel Booking |
- | 5.Add Room |
- | 6.Delete Room |
- | 7.Generate Reports |
- | 8.View Accounts |
- | 9.Delete Account |
- | 0.Exit |

5.3 User Menu Flow

[User Login Successful]

- | 1.Search Rooms |
- | 2.Book Room |
- | 3.Cancel Booking |
- | 0.Exit |

6. Booking & Cancellation Flow

1. User selects room → System checks All Rooms.txt
2. If available → Collect user info → Calculate cost → Save in Booked Rooms.txt
3. Update All Rooms.txt status → "Booked"
4. Cancellation reverses the process → "Available" in All Rooms.txt

7. Reporting Flow

- Reads All Rooms.txt and Booked Rooms.txt
- Calculates:
 - Total rooms, available, booked
 - Most/least booked room types
 - Revenue & occupancy
 - Registered users

8. Sample Output

-----Available Rooms-----

Room Number: 101

Room Type: Single

Price per Night: 5000

Description: One bed, suitable for 1 adult

Status: Available

Enter Room Number to Book: 101

-----Enter your Information-----

Enter Full Name: John Doe

Enter Contact Number: 03001234567

Enter CNIC: 12345-6789012-3

Enter total Nights: 3

Enter No of Guests: 1

Enter Check In date: 2025-08-16

Enter Check out dates: 2025-08-19

-----Booked Summary-----

Name: John Doe

Contact: 03001234567

CNIC: 12345-6789012-3

No of Guests: 1

Total Nights: 3

Room Number: 101

Room Type: Single

Price per Night: 5000

Status: Booked

Total Cost: 15000