

Architecture

Component Processes

```
graph TD;
    end_user_client["End User Client"] --> Server
    manager_client["Manager Client"] --> Server
    Server --> worker_1["Worker 1"]
    Server --> worker_2["Worker 2"]
    Server --> worker_3["Worker 3"]
    worker_1 --> map_reducer["Reducer"]
    worker_2 --> map_reducer["Reducer"]
    worker_3 --> map_reducer["Reducer"]
    map_reducer --> Server
```

Component Interactions

Operations

1. Manager to Server: Add a new room
2. End User to Server: Search for a room
3. End User to Server: Book a room

1. Manager to Server: Add a new room

```
sequenceDiagram;
    actor User

    User ->> ManagerClient: add a new room
    ManagerClient ->> Server: add a new room [json]
    Server ->> Server: generate id & hash
    Server ->> Worker(hash): save a room in memory
    Worker(hash) ->> Server: room stored
    Server ->> ManagerClient: done
    ManagerClient ->> User: ready (for next command)
```

2. End User to Server: Search for a room

```
sequenceDiagram;
    actor User

    User ->> EndUserClient: search for a room
    EndUserClient ->> Server: search criteria [json]
    Server ->> Server: generate job id & broadcast
```

```

par Server distributes search to Workers
    Server ->> Worker1: search with criteria, number of workers and
job id
    Server ->> Worker2: as above
    Server ->> Worker3: as above
end

par Reducer waits for workers to complete
    Worker1 ->> Reducer: Any search results with total workers and job
id
    Worker2 ->> Reducer: As above
    Worker3 ->> Reducer: As above
end

Reducer ->> Server: search results
Server ->> EndUserClient: search results
EndUserClient ->> User: formatted & paged search results

```

3. End User to Server: Book a room

```

sequenceDiagram;
actor User

User ->> EndUserClient: book room with room id
EndUserClient ->> Server: as above
Server ->> Server: Generate hash from room id
Server ->> Worker(hash): mark room as booked
Worker(hash) -->> Server: room booked
Server -->> EndUserClient: as above
EndUserClient -->> User: ready

```

Process Architecture

Client

```

classDiagram
Application --> TcpClient
Application --> Menu
Menu --> TcpClient

note "TcpClient connects to Server and transmits messages"

class TcpClient {
    connect()
    sendMessage(AddRoomRequest request)
}

```

```
class Menu {  
    show()  
    getChoice() (> TcpClient.sendMessage())  
}  
  
class Application {  
    TcpClient tcpClient  
    Menu menu  
}  
  
class AddRoomRequest {  
    // ... room fields  
}
```