**Tasks:**

1)Prepare more detailed description (ideally diagram) for the components interaction. Data flows and sequences (UML or any language of your choosing) will be highly appreciated.

2)Come up with a list of API endpoints with input/output contracts. Ideally to have a resource diagram as well.

3)Prepare sample json requests/responses for 1-2 endpoints of your choosing

4)Define user groups and access matrix (endpoints available for each user group)

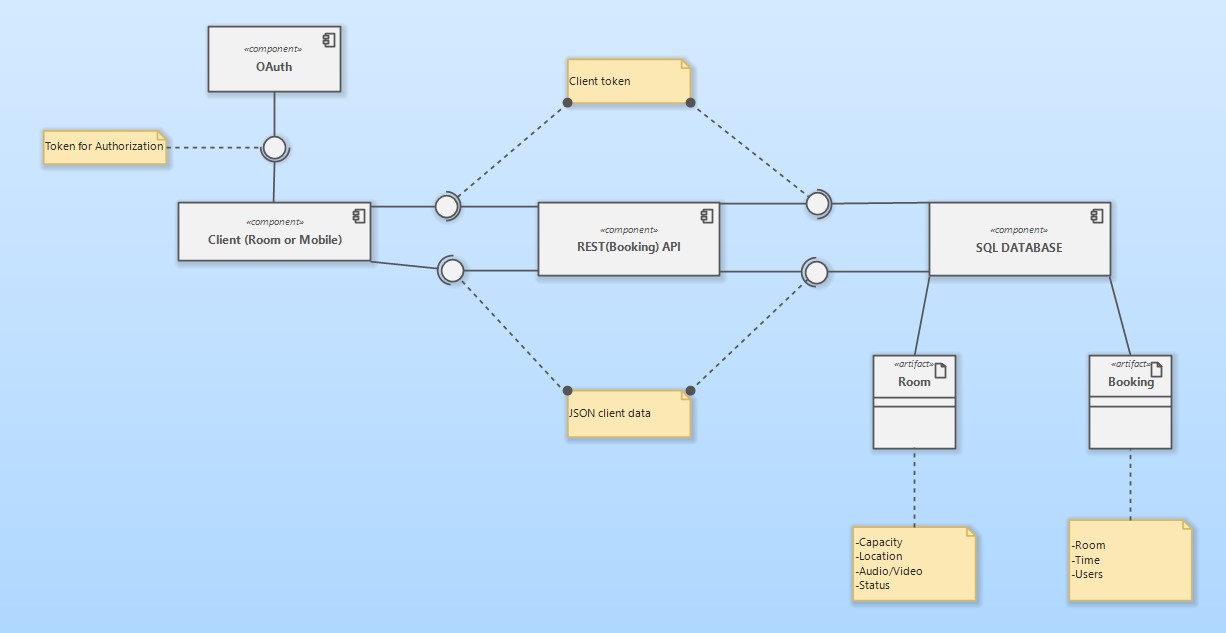
5)Define detailed functional requirements for one of the features of your choosing

6)Define non-functional requirements for the applications

7)Prepare high-level ERD diagram for the DB proposed\*

**Solution:**

1. Component diagram (Also you can find it in «https://github.com/Samkharadze/Align\_TEST/blob/main/Component%20diagram.jpg»)



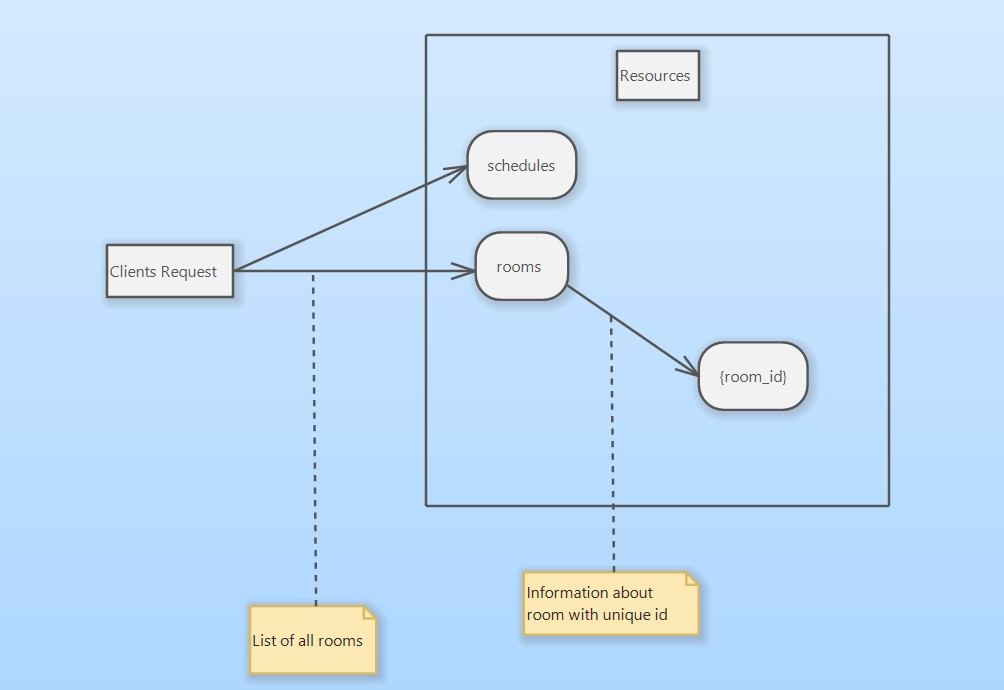
2)

GET /rooms/{room\_ID}-information about room with unique id: ID

GET /schedules-information about schedule

GET /rooms-list of all rooms

Resource diagram(Also you can find it in https://github.com/Samkharadze/Align\_TEST/blob/main/Resources%20diagram.jpg)



1. GET-request for client with id 143: /api/clients/143.json

GET-request for room with id 235: /api/rooms/235.json

4)Groups of users:  
 -Administrator

-User with access to information about him

-Any other users

|  |  |  |
| --- | --- | --- |
| Access | + | - |
| Viewing information about rooms | Administrator;User |  |
| Updating information about rooms | Administrator | Any other users |
| Viewing information about booking | Administrator;User |  |
| Updating information about booking | Administrator; User calling the API to update his own booking information | Any other users |
| Viewing information about user | Administrator; User calling the API to see information about himself | Any other user |
| Updating information about user | Administrator; User calling the API to update his own information | Any other users |

5) Functional requirements for the function of booking:

-OAuth mechanism must be used for authorization

- The system must provide the user with a suitable means of viewing information about the reservation itself (number, time, etc.)

-The system should provide the user with information about conflicts when booking

-The system should provide the user with the opportunity to update information about their reservation (book a room, etc.)

-Interaction with REST API

-Reservation data must be stored in SQL DB

- All necessary data for clients should be provided in the form of json entities

6)Non-functional requirments:

-The system must be available for users at any time

- The system must be available to the user through a mobile application or a special screen

-The user interface must be accessible and provide all the necessary information for a comfortable room reservation

-The system must be reliable: store the data of all users and information about their reservations