What functional components would suggest implementing based on the visual view of the application interface?

Functional Components will include:

**Devices**

1. Home page will show a list of all devices with their status and the icon can be mapped to each device types e.g. iOS, Android, Desktop
2. Device info pages will show the device info such as status, temperature, usage which can be how often the status changes grouped over time (e.g. daily/weekly/monthly)
3. Device info pages will also have a section to show related devices which could be devices that has the same device type with the current device being viewed.
4. There will be a page to register a new device with details such as device name, current status, current temperature.
5. There will also be a functionality to edit a device with the updated details.
6. There will also be a functionality to delete a device.

**Device Types**

1. There will be functionality to create a new device type.
2. There will be a page to list all device types.
3. There will be functionality to update a device type.
4. There will be a functionality to delete a device type.
5. There will be a functionality to get a device type and view all its details.

**Device Status**

1. There will be functionality to create a new device status.
2. There will be a page to list all device statuses.
3. There will be functionality to update a device status.
4. There will be a functionality to delete a device status.
5. There will be a functionality to get a device status and view all its details.

**Device Status Log**

1. There will be a page to add a device status change. This is used to log a change of the status of a device. E.g. from available to office.
2. There will be a page to view all changes in device status of all devices over time.
3. There will be functionality to update a device status log.
4. There will be a functionality to delete a device status log.

**Token**

1. There will be a functionality to handle user authentication and authorization.
2. All API endpoints are protected with JWT Authorization except for login and user registration.

**User**

1. There will be a functionality to register a new user.

What are the API endpoints needed by the application and example routes for each endpoint?

Devices:

1. Add Device [POST] {{base\_url}}/v1/devices/addasync
2. Get Devices [GET] [Paginated, includes search term] {{base\_url}}/v1/devices/getpagedasync/pagenumber/pagesize/query
3. Get a Device by ID [GET] {{base\_url}}/v1/devices/getasync/id
4. Get Similar Devices BY Device ID [GET] {{base\_url}}/v1/devices/getsimilardevices/id
5. Update a Device [PUT] {{base\_url}}/v1/devices/updateasync/id
6. Delete a Device [DELETE] {{base\_url}}/v1/devices/deleteasync/id

Device Types:

1. Add Device Type [POST] {{base\_url}}/v1/devicetype/addasync
2. Get Devices Type [GET] [Paginated, includes search term] {{base\_url}}/v1/devicetype/getpagedasync/pagenumber/pagesize/query
3. Get a Device Type by ID [GET] {{base\_url}}/v1/devicetype/getasync/id
4. Update a Device Type [PUT] {{base\_url}}/v1/devicetype/updateasync/id
5. Delete a Device Type [DELETE] {{base\_url}}/v1/devicetype/deleteasync/id

Device Status:

1. Add a Device Status Type [POST] {{base\_url}}/v1/devicestatus/addasync
2. Get Devices Status Type [GET] [Paginated, includes search term] {{base\_url}}/v1/devicestatus/getpagedasync/pagenumber/pagesize/query
3. Get a Device Status Type by ID [GET] {{base\_url}}/v1/devicestatus/getasync/id
4. Update a Device Status Type [PUT] {{base\_url}}/v1/devicestatus/updateasync/id
5. Delete a Device Status Type [DELETE] {{base\_url}}/v1/devicestatus/deleteasync/id

Device Status Log:

1. Add a Device Status Activity Log [POST] {{base\_url}}/v1/devicestatuslog/addasync
2. Get Devices Status Activity Log [GET] [Paginated, includes search term] {{base\_url}}/v1/devicestatuslog/getpagedasync/pagenumber/pagesize/query
3. Get a Device Status Activity Log by ID [GET] {{base\_url}}/v1/devicestatuslog/getasync/id
4. Get all Device Status Activity Log [POST] {{base\_url}}/v1/devicestatuslog/getalldevicestatusactivitylog
5. Get a Device Status Activity Log [POST] {{base\_url}}/v1/devicestatuslog/getdevicestatusactivitylog
6. Update a Device Status Activity Log [PUT] {{base\_url}}/v1/devicestatuslog/updateasync/id
7. Delete a Device Status Activity Log [DELETE] {{base\_url}}/v1/devicestatuslog/deleteasync/id

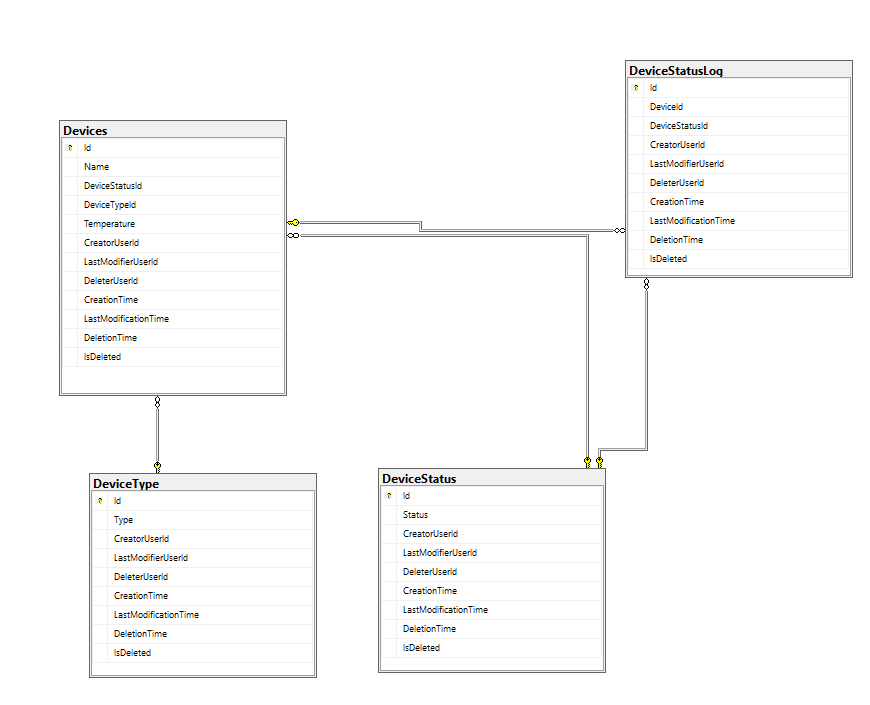
Token:

1. Fetch Token by supplying username and password [POST] {{base\_url}}/v1/token/auth
2. Refresh Token by supplying the access token and refresh token gotten from (1) [POST] {{base\_url}}/v1/token/RefreshToken
3. Logout user to record the log out time of a user [PUT] {{base\_url}}/v1/token/logout/{userId}

User:

1. Register a new user [POST] {{base\_url}}/v1/user/registeruserasync

How would you model the necessary DB tables between users, devices, and available device operations?

****

Devices Types has its id as a foreign key on device table.

Device Status has its id as a foreign key on device table.

Device Table has its id as a foreign key on device status log table.

Device Status has its id as a foreign key on device status log table.

Users are not tied to devices as there was no indication it should.

Device Table has a device status id which can be used to get a list of all available devices. i.e. devices with device status ‘available’.

PART II

1. I prefer style 2, as it gives a more holistic view of all devices.

In terms of improvements, I would my acceptance criteria to cover

1. Updating a device detail.
2. Deleting/Deactivating a device detail.
3. There are audit trails on my application.
4. Ensuring my entire application is protected with some level of authorization.

2. Tests should cover getting all devices as a paginated list to make it faster, including the device status, device type. Device types can be mapped to specific icons on the dashboard e.g ios: iPhone-icons, android: android-icons etc.

3. Tests should ensure getting device details by device ID returns all necessary details.

Tests should ensure getting all devices should not load all devices at once and it is paginated.

Tests should ensure getting device details by device ID returns Device Status Activity Log. This shows the device status change logs.

Tests should ensure functionality to return all device status activity logs which can show which devices have the highest changes in their status. This can be used to populate the dashboard for most used devices