System Documentation of
Patient Monitoring and Control System

Author: Axel Afonso Blanco
Eldar Guliyev
Qingqin Hua

1.Road-map

This documentation describes the architecture overview of the Patient Monitoring and Control System. It contains the overview of the system[chapter 2] and architecture viewpoints[chapter 3].

Stakeholders can find detailed information about the system and their requirements in system overview[chapter 2], where they can get an overview of what is the purpose of the system[2.1], and what is considered when developing the system[2.2].

Developers may be interested in architectural viewpoints[chapter 3], where details of different architectural viewpoints are explained. Developers may also take a look at section 2.3, where a list of requirements are listed from the concerns of the stakeholder.

The architecture viewpoints used in this documentation are module viewpoint [3.2] and component & connector viewpoint[3.3]. The module viewpoint describes how the system is decomposed into different modules and their relationships. In the part of the module viewpoint, we describe the overview of the modules by decomposition viewpoint[3.2.1] and usage viewpoint[3.2.2]. In the component & connector viewpoint, the running behavior of the system is discussed.

2. System Overview

2.1. Purpose of the system

The Patient Monitoring and Control System is a standalone system in Hospital Management. It is used for monitoring the health of the patient and controlling the location of patients.

With this system, doctors and nurses working in the hospital are able to monitor the information of the patients in each department. The system provides up to date patients health status and their location so that the doctors and nurses can track their patients immediately. The system also provides relocate functionality so that the doctors and nurses can move the patient to another bed or room. The system is accessible from the employee's mobile device using the provided account.

2.2. Stakeholders and their concerns

Patients - Patients want to have good attention in the hospital.

Medical staff - Hospital workers such as doctors,nurses use systems for getting/setting patients allocation, updating patients data and controlling real time health status of patients.

Administration - Department managers and administrators who control the medical staff's data, patient data, rooms and departments in the hospital.

Developer - Developers can maintain performance of the code,optimize systems workload, fix appearing bugs and improve operations processing time.

2.3. Requirements from stakeholder

- Functional requirements
    - The system must read information from the central database regarding departments, rooms, patients and doctors.
    - The system must monitor the number of rooms per department.
    - The system admin must be able to change the number of rooms per department.
    - The system must monitor the number of patients per department.
    - The system must be able to add and remove patients when they are added to or removed from the central database.
    - The system must monitor patient allocation in rooms.
    - The system must read heart rate and temperature information of each patient from medical devices through their API.
    - Medical staff must be able to read information from the system regarding patient allocation, and heart rate and temperature monitoring.
    - Medical staff must be able to relocate patients from one room to another one.
    - All interactions between medical staff and the system must be made through their mobile devices.

- Quality(Non-functional) requirements
    - Performance:
    - The system must handle 5 or 6 departments with 20 to 50 rooms per department and 1 or 2 patients per room.
    - Retrieving information from the medical devices must take seconds.
    - Retrieving any other information from the system must take less than a minute.

    - Security:
    - The system must only be accessible from inside the hospital.
    - Medical staff must log in before accessing any information.
    - Only medical staff and the system admin can read information from the system.
    - Medical staff can only read information from their department.
    - Medical staff can only relocate patients from one room in their own department to another room in their own department.

    - Modifiable:
    - Features will not change.
    - The API of the medical devices may change.

3. Views documentation

In this chapter, we will discuss the system architecture by using different architectural views.

The architectural views used here are, decomposition viewpoint[3.2.1], which is helpful for understanding how the system is structured by decomposing it into code units.

In connection with the decomposition viewpoint, the usage viewpoint[3.2.2] is used to explain the relationships between each module.

The Component & connector viewpoint is explained in section[3.3], it is used to show run-time elements and relationships between the elements. From this viewpoint, you can see how the system is structured in running time.

3.2 Module Viewpoints

The decomposition viewpoint and usage viewpoint are provided below to represent the architecture of the system.

3.2.1 Decomposition Viewpoint

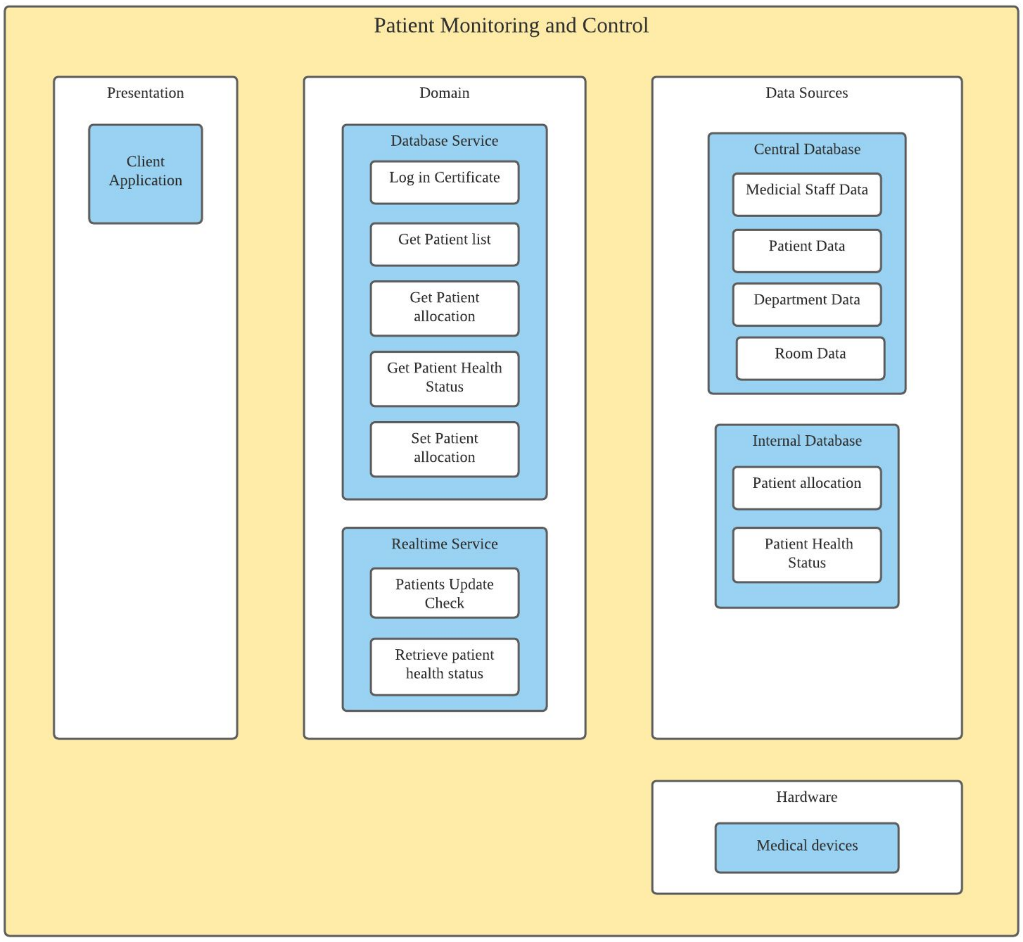

Diagram 1: The decomposition viewpoint of the system

The Patient Monitoring and Control System are decomposed into four layers, from left to right, they are the Presentation layer, Domain layer, Data Sources layer and Hardware Layer. Each of the layers is explained in detail in the next paragraph.

The hardware layer provides the functionality to retrieve the information from the medical device.

The domain layer provides the functions to connect the presentation layer and data sources layer. It is decomposed into two modules:

Database services module, which is mainly for getting data from the *Central Database* and *Internal Database* in the Data Sources layer. The main functionalities in this module are:

- Log in Certificate: Get the password and username of required medical staff from *Medical Staff Data* in *Central Database,* check if the log-in certificate can pass.
- Get Patient list: Get all the patient’s information from *Patient Data* in the *Central Database.*
- Get Patient allocation: Get the allocations of patients from *Patient allocation* in *Internal Database.*
- Get Patient Health Status: Get the health status of patients, such as heartbeats and temperature from *Patient Health Status* in the *Internal Database.*
- Set Patient allocation: Relocate the patient, update the data to *Patient allocation* in *Internal Database.*

Realtime Service Module, this module is running in real-time in the background of the system. This module is standalone, so it doesn’t need to be called from *Client Application*. Real-time presents that this module will be called every 5 seconds in the system. There are two main functionalities in this module:

- Patient Update Check: Check if the Patient list from *Central Database* has been changed. If so, add or delete the corresponding patient in the *Internal Database.*
- Retrieve patient health status: Read the health status of each patient from a medical device, write it to the *Patient Health Status* in *Internal Database.*

The Data Sources layer is responsible for sending the data from the domain layer to the database where the data is persisted and also is used for extracting the data from the database. It is decomposed into two modules:

The central database module is a module where all hospitals’ main data is stored.It consists of several parts:

- *Medical staff data i*s the personal data of each hospital worker such as:
  - Name
  - Surname
  - Specialization

- Position(doctor,nurse,ambulance paramedic,administrative manager or help desk receptionist).

- *Patient data* stored in a central database saves all information about the patient's name, surname, and age.

- *Department data* contains the whole information about each department of the hospital: - Name of the department
  - List of all medical staff working in the department
  - Total number of current patients in the department
  - Total number of rooms per department
  - Location

- *Room data* stores the data about each room in some particular department and consists of such a set of parameters:
  - The number of room
  - Number of patients in a room
  - List of patients located in a room
  - Location of a room (floor, department)

The internal database module is a second data source module which stores real-time changing data:

- *Patient allocation:* saves the information about the current patient's location such as:
  - Name of the department.
  - Number of the room.
- *Patient health status:* constantly updates and stores the real-time data coming from the patient's monitoring medical device:
  - Heart rate
  - Temperature information

The Presentation layer

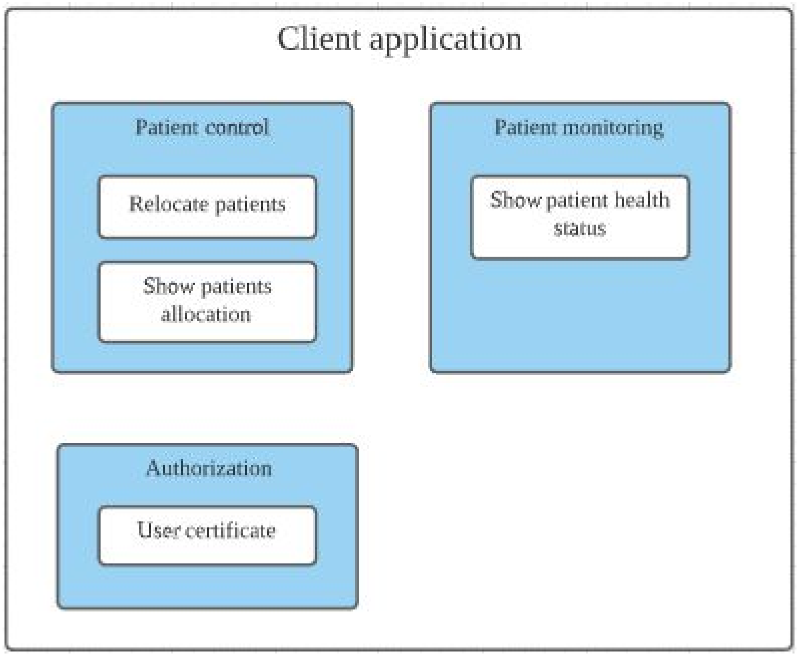

Diagram 2: Decomposition of the Client Application

The presentation layer consists of a client application that retrieves information from the domain layer and presents it to the medical staff and provides an interface for the medical staff to interact with the system. It is decomposed into three modules:

Patient control: Manages information regarding patient allocation. It is divided in two submodules:

- Show patients allocation, which shows which patient is in each room.
- Relocate patients, which provides an interface for the medical staff to relocate patients to another room.

Patient monitoring: It is composed by one submodule, Show patient health status, which provides temperature and health information for each patient.

Authorization: takes care of the security aspects of the application. It is composed of only one submodule:

- User certificate, which manages the login into the application.

3.2.2 Usage Viewpoint

In this section, we will describe usage between each of the code units.

The arrows here are representing the connection between each code unit.
The colors of the arrows are used to distinguish between different modules. Green lines are pointed to the Internal Database. Yellow lines are pointing to the Central Database. Black lines are pointing to Database Service. The red lines are pointed to the Hardware layer.

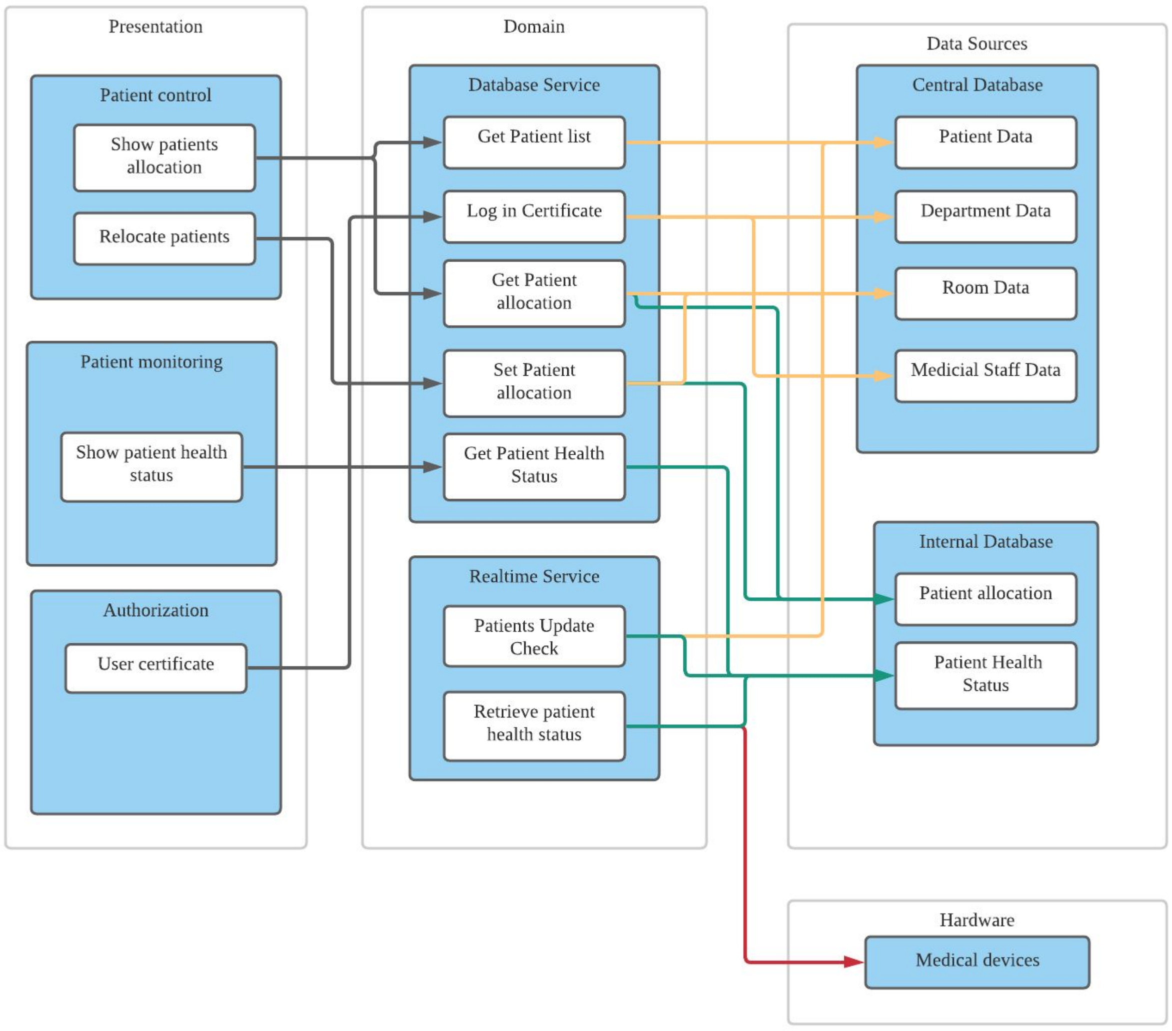

Diagram 3: Usage viewpoint of the system

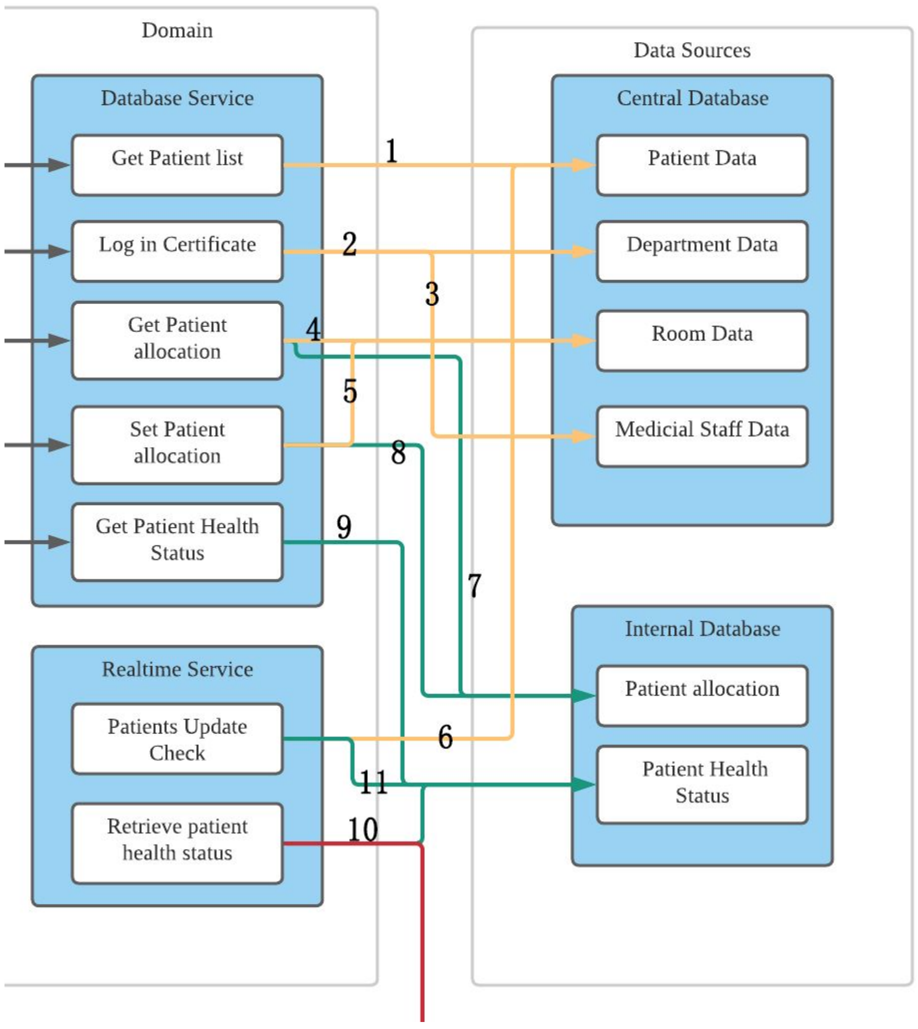

Diagram 4:Usage between Domain Layer and Data Sources Layer

Arrows to Central Database in Diagram 4:

All **yellow arrows** represent the database service's operations for retrieving and updating data from the central database.

1) Get a patient list from all the patient's data stored in the central database.
2) Getting the department data of logged in medical staff users.
3) Getting the name and surname of a logged-in medical staff user.
4) Getting the room data in which the patient is currently located.
5) Setting the room data in which the patient is currently located.
6) Getting updated information about the patient's data from the central database.

Arrows to Internal Database in Diagram 4:

All **green arrows** represent the database service's operations for retrieving and updating data from the internal database.

7) Get the patient allocation from Internal Database
8) Set new patient allocation to Internal Database
9) Get patient health status from Internal Database

10) Read and write patient heal status from internal database.
11) Update the patient allocation whenever the Patient UpdateCheck finds the change in the patient list.

Moreover, the functionalities in RealTime Service are run standalone in real-time.

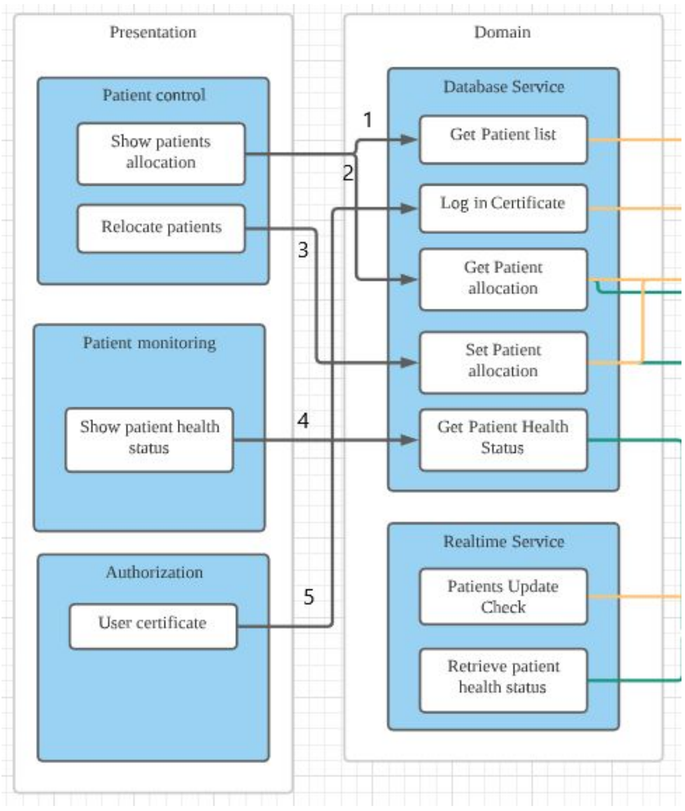

Diagram 5: Usage between Presentation Layer and Data Sources Layer

Arrows to Domain Layer in Diagram 5:

All **black arrows** represent the Presentation layer's operations for getting data and performing actions from the domain layer

1) Get the patient list for the doctor's department.
2) Get the room list for the doctor's department.
3) Sending the patient's name and its new room.
4) Getting the heart rate and the temperature.
5) Login in with the doctor's name and password.

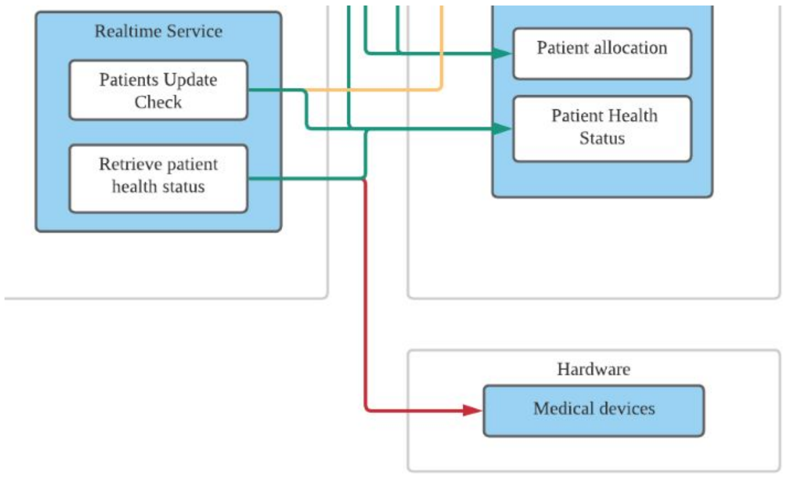

Diagram 6: Usage between Realtime Service and Hardware

Arrows to Hardware Layer in Diagram 6:
The red arrow represents the operations from Realtime Service for getting data from the Medical devices. This operation runs in realtime, meaning it will refresh the data every second.

3.3 Component&Connector viewpoint

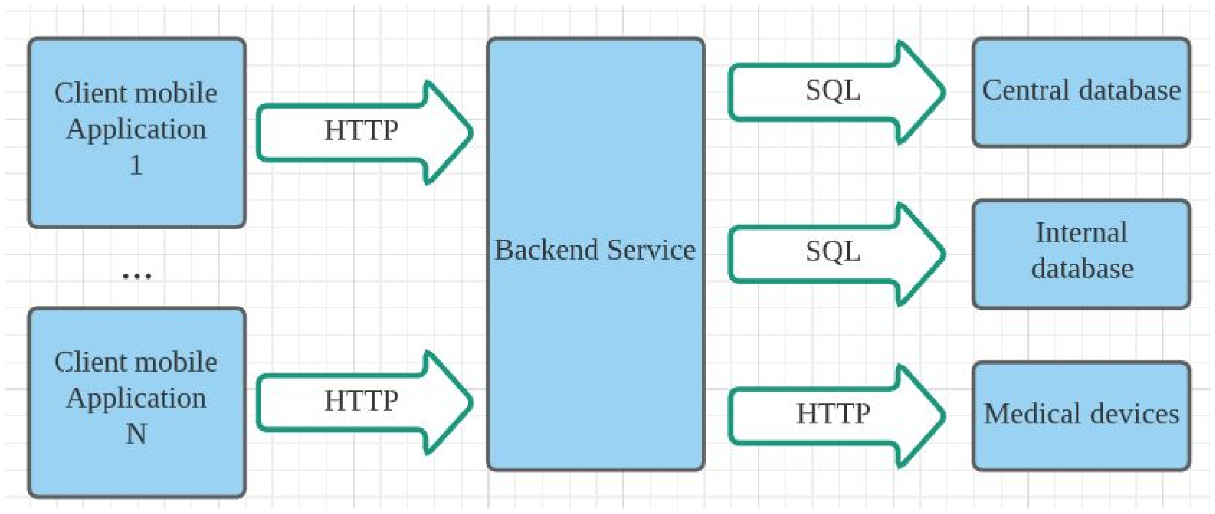

Diagram 7: C&C diagram of the system

Components:

- The **client application** component runs in the medical staff's mobile devices, so there can be multiple instances running at the same time.This component runs the code inside the client application module (decomposition viewpoint).

- The **Backend Service** component runs in a server, so it is only instantiated once.It runs all the code inside the Domain Layer (decomposition viewpoint).

- The **Central database** component is given by the hospital, and it runs its only instance in a server independently from the rest of the system.

- The **Internal database** component is only instantiated once, and stores the information contained in the internal database module (decomposition viewpoint).

- Finally, the **Medical devices** component consists of multiple medical devices spread across the hospital rooms, so it is instantiated multiple times.

Connectors:

- **Client application** components communicate with the **Backend Service** component through HTTP requests.

- The **Backend Service** component communicates with the **Central Database** component through SQL, thereby accessing the data in the database directly.

- The **Backend Service** component communicates with the **Internal Database** component through SQL, thereby accessing and modifying the data in the database directly.

- The **Backend Service** component communicates with the **Medical devices** components through HTTP.