

**Enterprise Technology – Mobile Platforms  
(MAPD713)**

**Project: Patient Data REST API**

**Submitted to: Prof. Rania Arbash**

**By:**

<b>Name</b>	<b>Id</b>
Bhargav Borse	301278352
Khanjan Dave	301307330

## 1. Introduction

The effective management of patient clinical data is not only necessary, but also a crucial component of offering high-quality healthcare services in today's quickly changing healthcare environment. In an effort to improve patient care and lighten administrative duties, hospitals and other medical facilities are increasingly turning to digital solutions to streamline and optimize their workflows. A substantial step towards attaining this objective has been made with the Patient Data REST API initiative.

### 1.1. Background

To manage patient information in the past, healthcare providers used paper-based records and fragmented systems. This method has frequently led to waste, delays, and the potential for mistakes in patient care. Electronic health records (EHRs) and modern technology have caused a paradigm change in the healthcare sector. The difficulty, though, is in successfully utilizing digital tools' power to offer easy access to patient data while protecting data security and privacy.

### 1.2. Scope

The design, creation, and deployment of a RESTful API service suited to the requirements of healthcare professionals, including doctors and nurses, are all included in the project's scope. The API enables the addition and retrieval of comprehensive clinical test results as well as general patient information. A critical patient identification tool is also included, which analyzes health indicators to identify patients with critical conditions.

The goal of the project is to strike a balance between scalability, data security, and user-friendliness. It seeks to provide a complete answer to the problems healthcare providers confront in effectively handling patient data.

## 2. Requirement analysis

### A. User Authentication and Authorization:

- The ability for users (healthcare providers) to securely log in is required.
- Different user roles (such as doctors and nurses) ought to have different access levels and permissions.

### B. Add Patient Information:

- General patient data, such as name, age, gender, contact information, and medical history, should be added by healthcare professionals.
- This information needs to be securely validated and stored by the system.

**C. View Patient Information:**

- Users should be able to do a search using patient identifiers and read the general information for a particular patient.

**D. List All Patients Information:**

- A list of each patient's general information, which can be paginated for convenience, should be provided via the API.

**E. Add Clinical Tests for a Patient:**

- The ability to input clinical test information for a single patient, including the date, time, type of data, and readings or values, is required.
- The data should be verified by the system, and it should be linked to the appropriate patient.

**F. View Clinical Tests of a Patient:**

- Users ought to have access to and be able to examine comprehensive clinical test data for a particular patient.
- The information must be structured and presented clearly.

**G. Delete Patient Information:**

- In order to comply with data privacy laws, healthcare providers should have the flexibility to erase patient information as needed.
- The deletion procedure must be secure and permanent.

**H. Delete Clinical Test Data:**

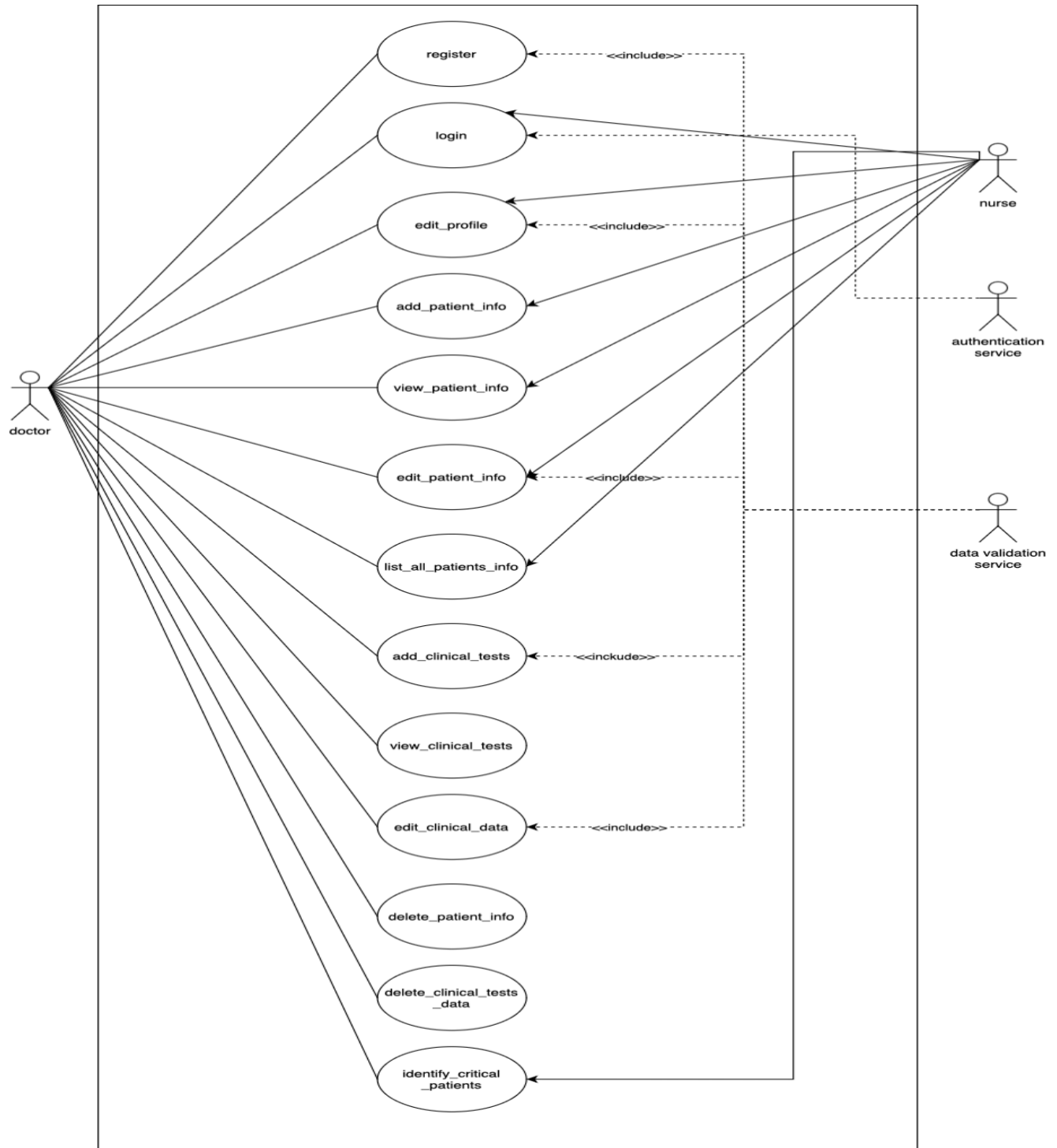
- When appropriate, clinical test data for a single patient should be erasable by healthcare providers.
- Deletion must be carried out safely and with a log for auditing.

**I. Identify critical patients:**

- A component that allows the system to recognize patients in critical condition based on particular health parameter criteria, like blood pressure, should be included.
- Reliable and prompt identification of critical patients is required.

### 3. Use case Diagram

- All diagrams link:  
(<https://github.com/BhargavBorse/CentennialCollege/tree/9c32bba9806e9565c39b87d40f7eddeac9fa18c7/Enterprise/Project%20files/Diagrams>)



## A. Use case descriptions

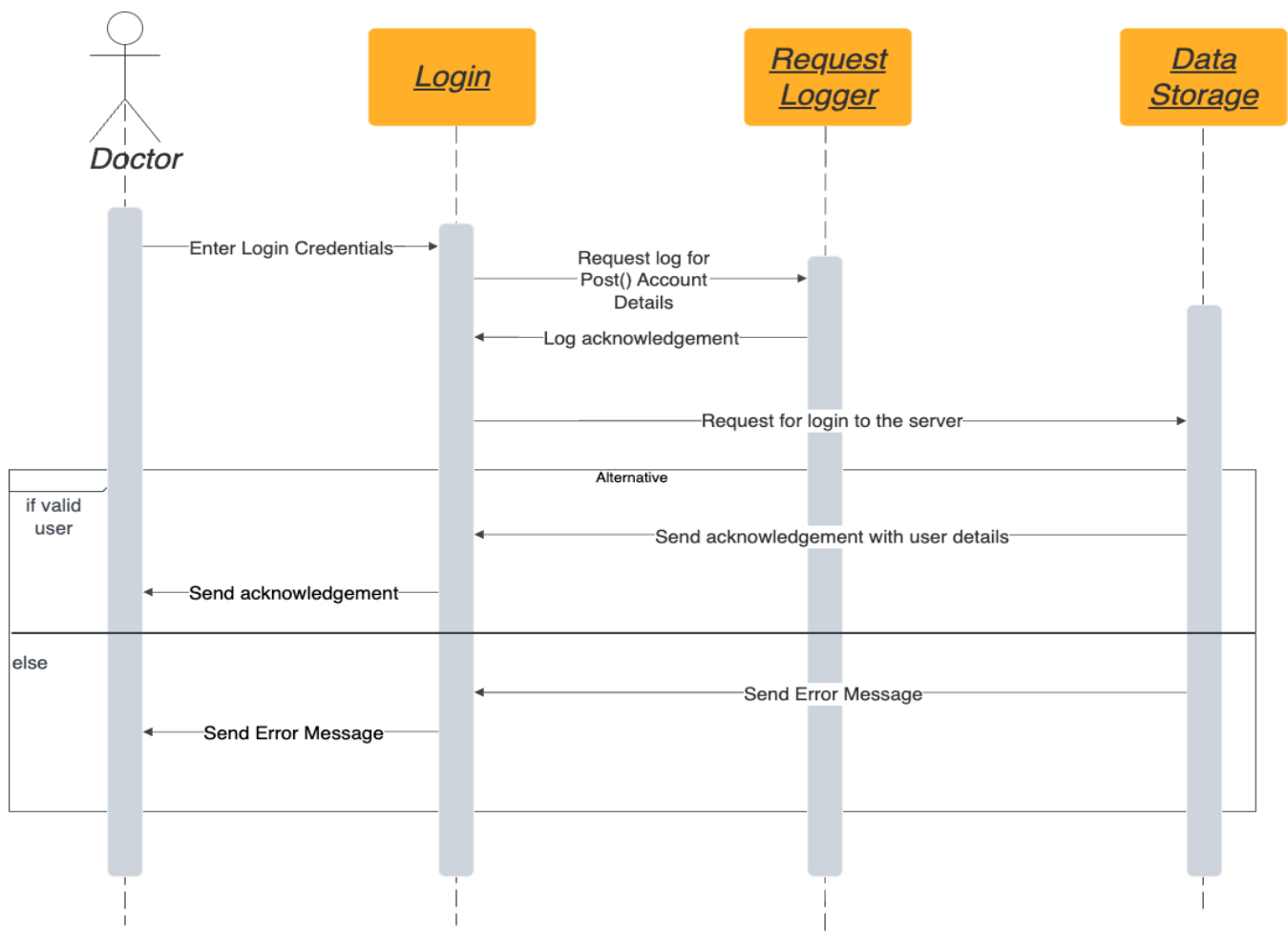
- **Login:**
  - **Actor:** Doctor, Nurse
  - **Description:** Allows healthcare providers to log into the system securely using their credentials.
- **Add patient info:**
  - **Actor:** Doctor, Nurse
  - **Description:** Enables healthcare providers to add general patient information, including name, age, gender, contact details, and medical history.
  - **Relationship:** <<include>> Data Validation: Includes data validation checks through the "Data Validation Service" to ensure correctness and data type compliance.
- **View patient info:**
  - **Actor:** Doctor, Nurse
  - **Description:** Allows healthcare providers to view the general information of a specific patient by searching using patient identifiers.
- **List all patient info:**
  - **Actor:** Doctor, Nurse
  - **Description:** Provides a list of all patients' general information for quick access. This can be paginated for easier navigation.
- **Add clinical tests:**
  - **Actor:** Doctor
  - **Description:** Permits doctor to add clinical test data for a specific patient, including date/time, type of data, and readings/values.
  - **Relationship:** <<include>> Data Validation: Includes data validation checks through the "Data Validation Service" to ensure correctness and data type compliance.
- **View clinical tests:**
  - **Actor:** Doctor
  - **Description:** Enables doctor to retrieve and review detailed clinical test data for a specific patient.
- **Delete patient info:**
  - **Actor:** Doctor, Nurse
  - **Description:** Allows healthcare providers to delete patient information if required. The deletion process should be secure and logged for auditing.

- **Delete clinical test data:**
  - **Actor:** Doctor
  - **Description:** Permits doctor to delete clinical test data for a specific patient when necessary. Deletion should be done securely and logged for auditing.
- **Identify critical patients:**
  - **Actor:** Doctor, Nurse
  - **Description:** The system has a feature to identify patients in critical condition based on specific health parameter criteria, such as blood pressure. Critical patient identification should be reliable and immediate.

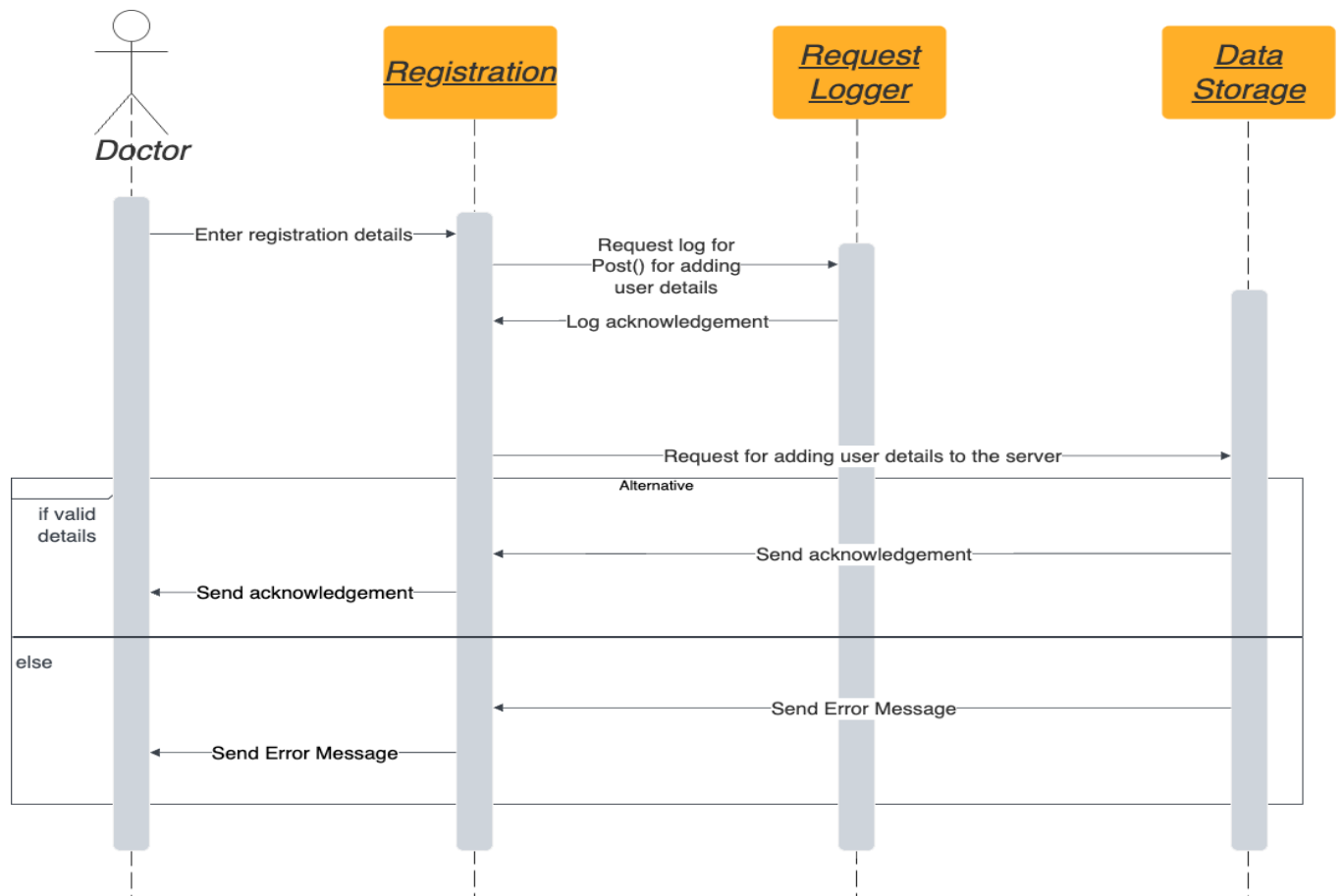
## 4. Sequence Diagram

- All diagrams link:  
(<https://github.com/BhargavBorse/CentennialCollege/tree/9c32bba9806e9565c39b87d40f7eddeac9fa18c7/Enterprise/Project%20files/Diagrams>)

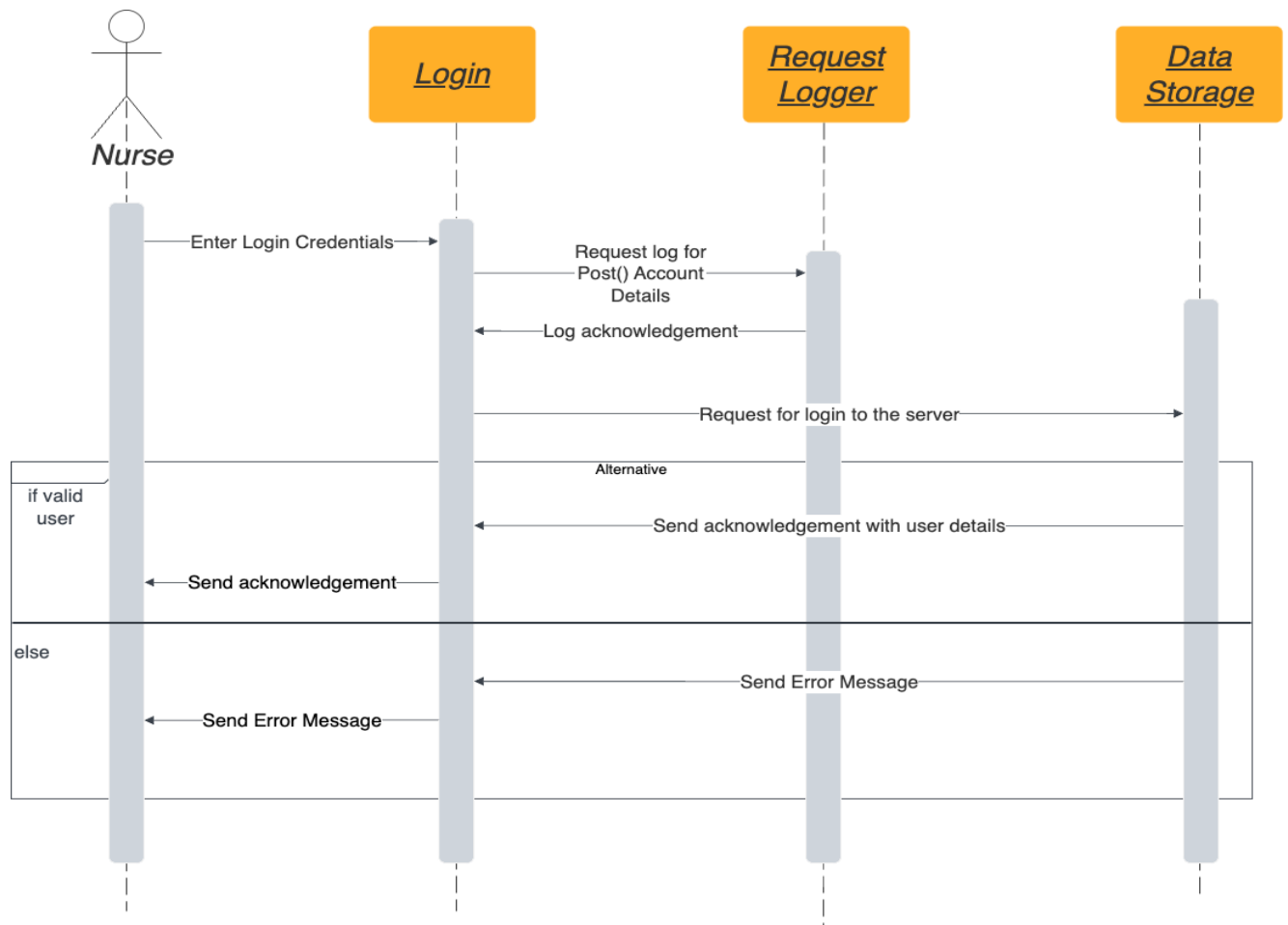
### a) Doctor Login



b) Doctor registration

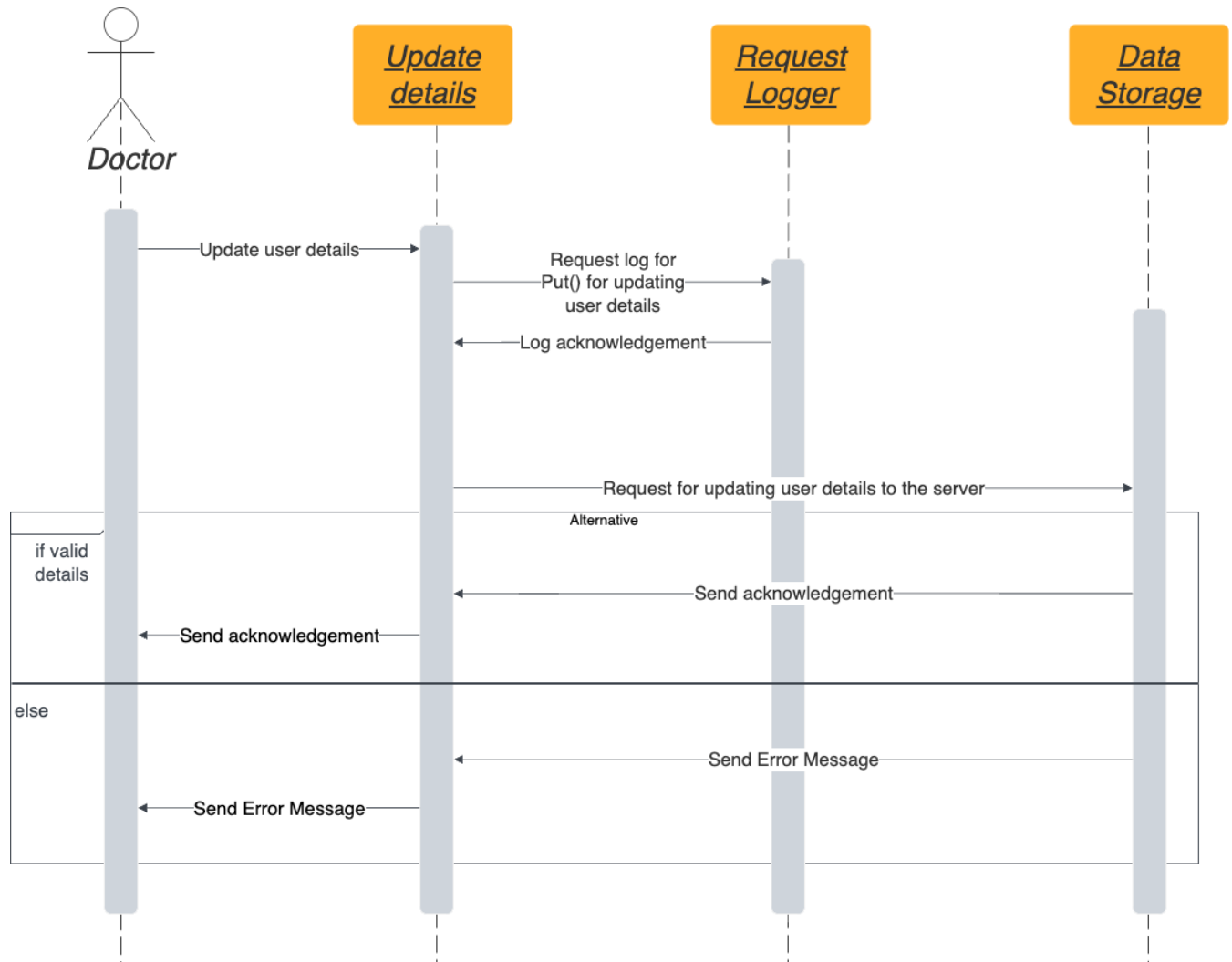


c) Nurse login

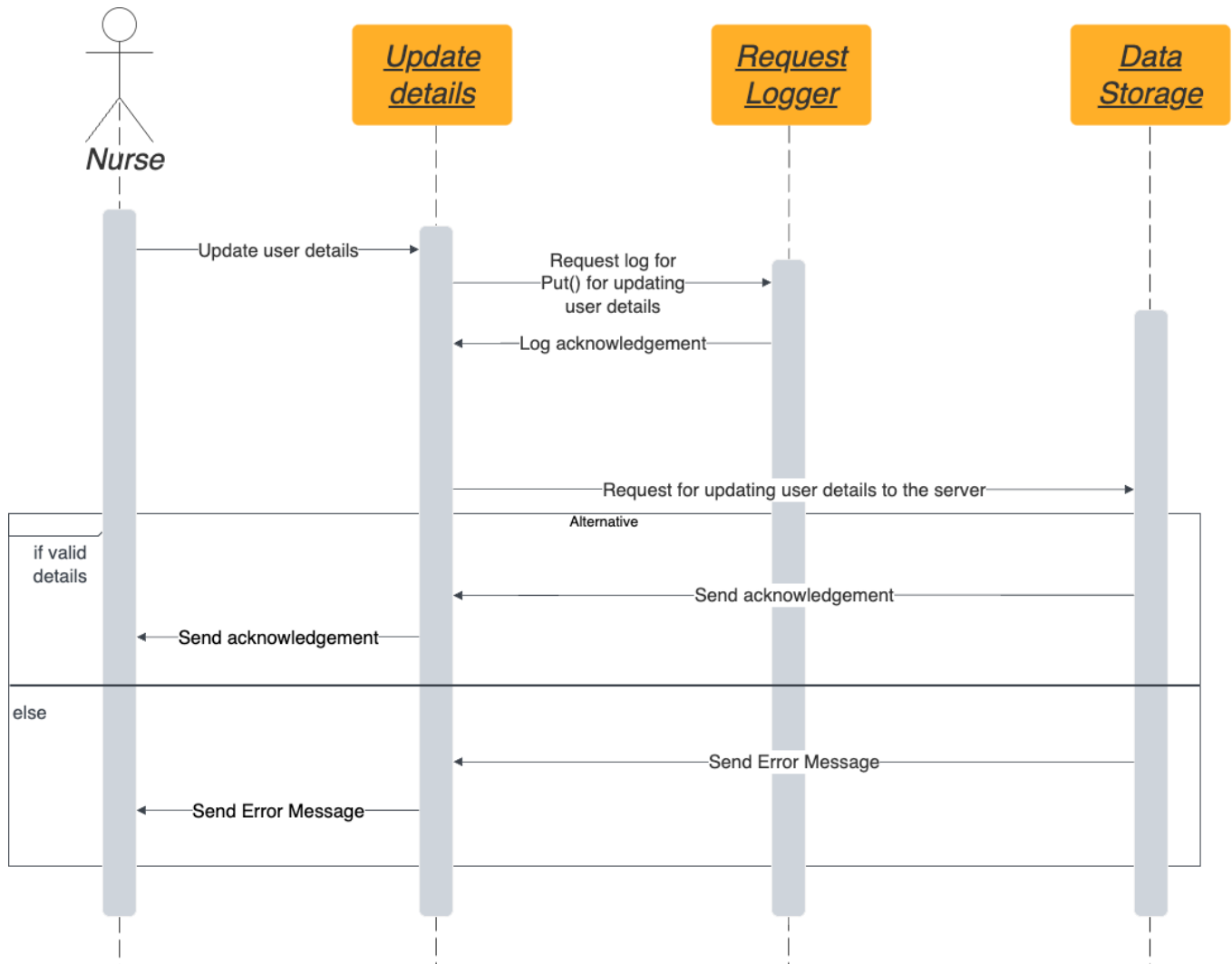




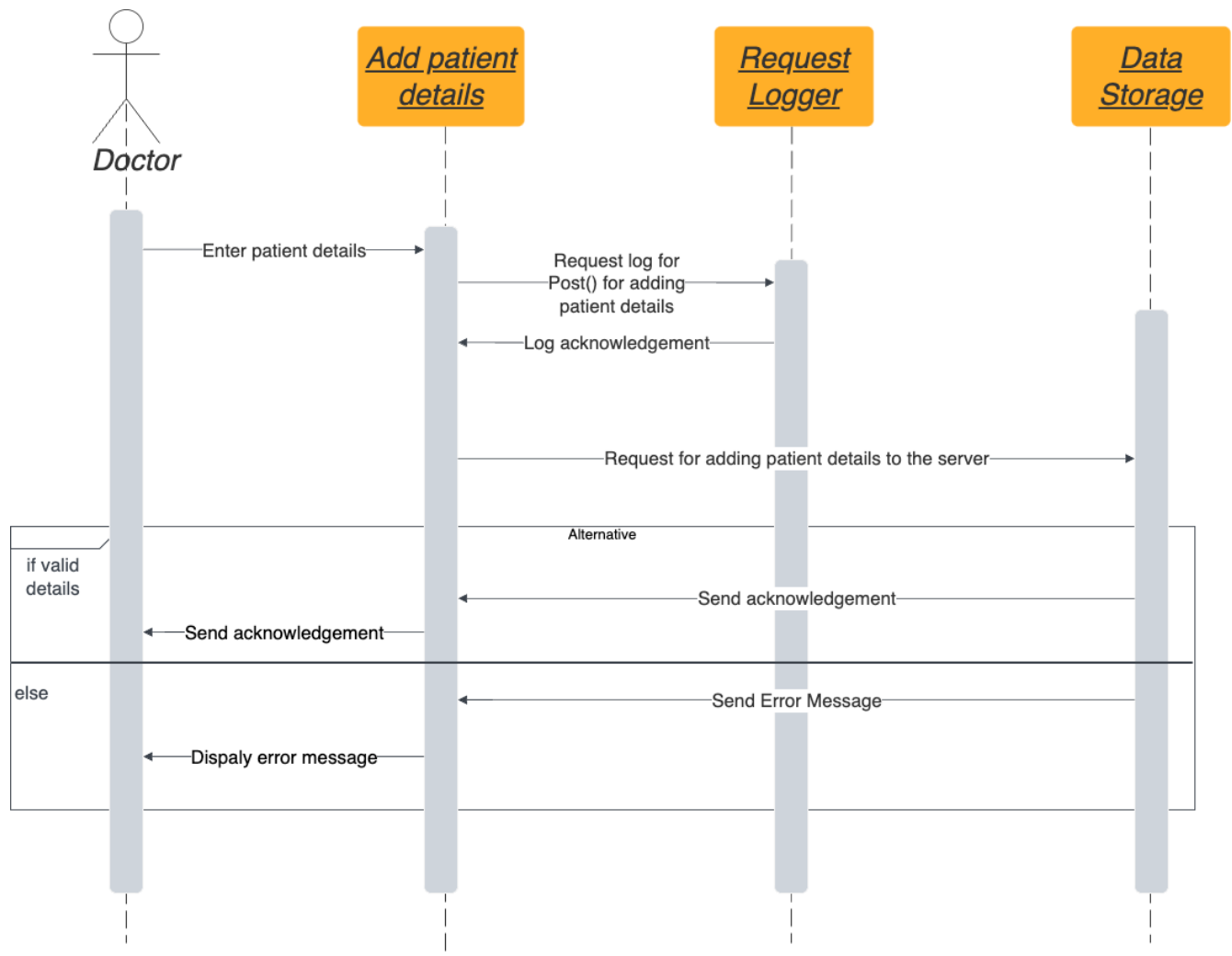
d) Doctor details update



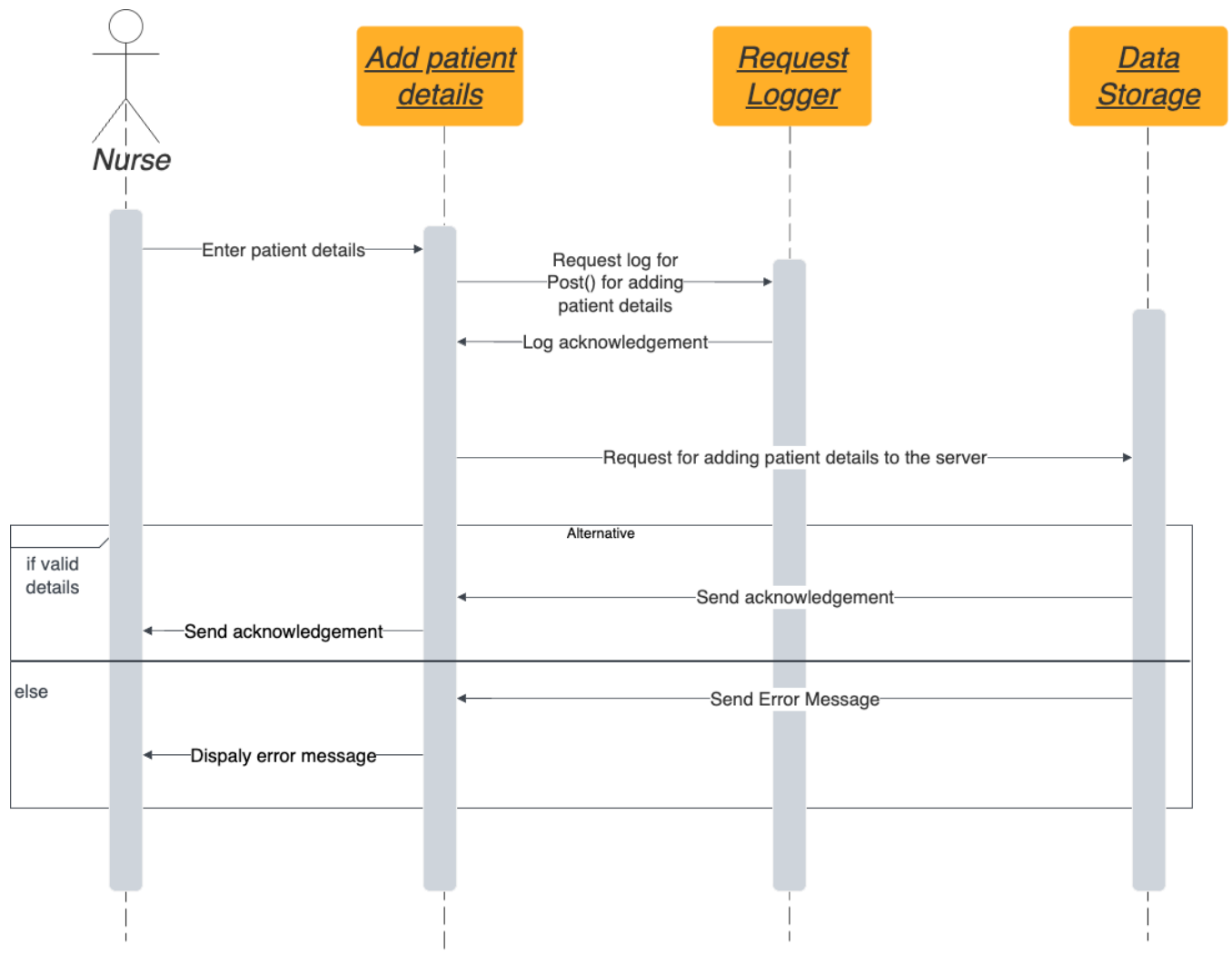
e) Nurse details update



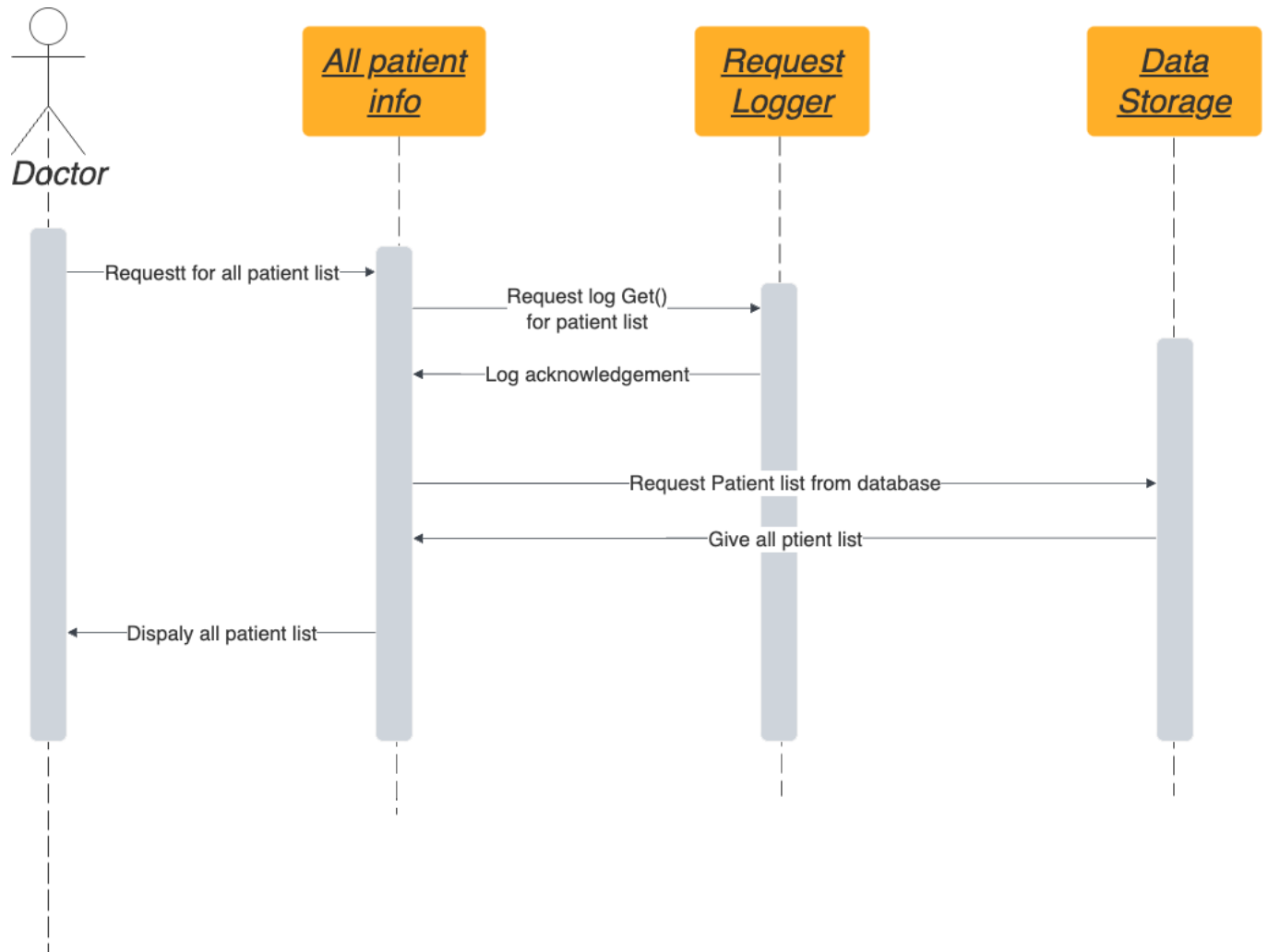
f) Doctor – add patients details



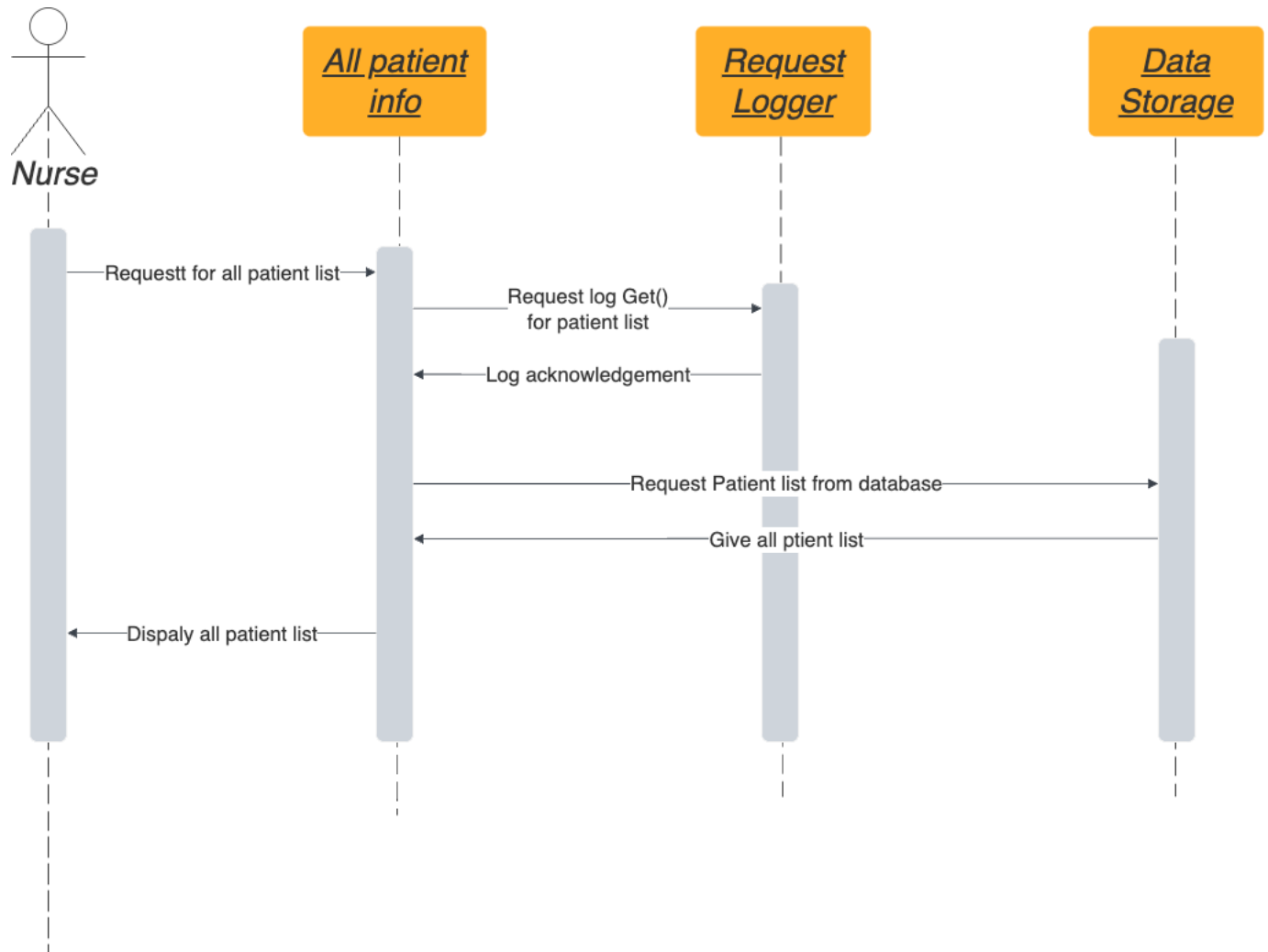
g) Nurse – add patients details



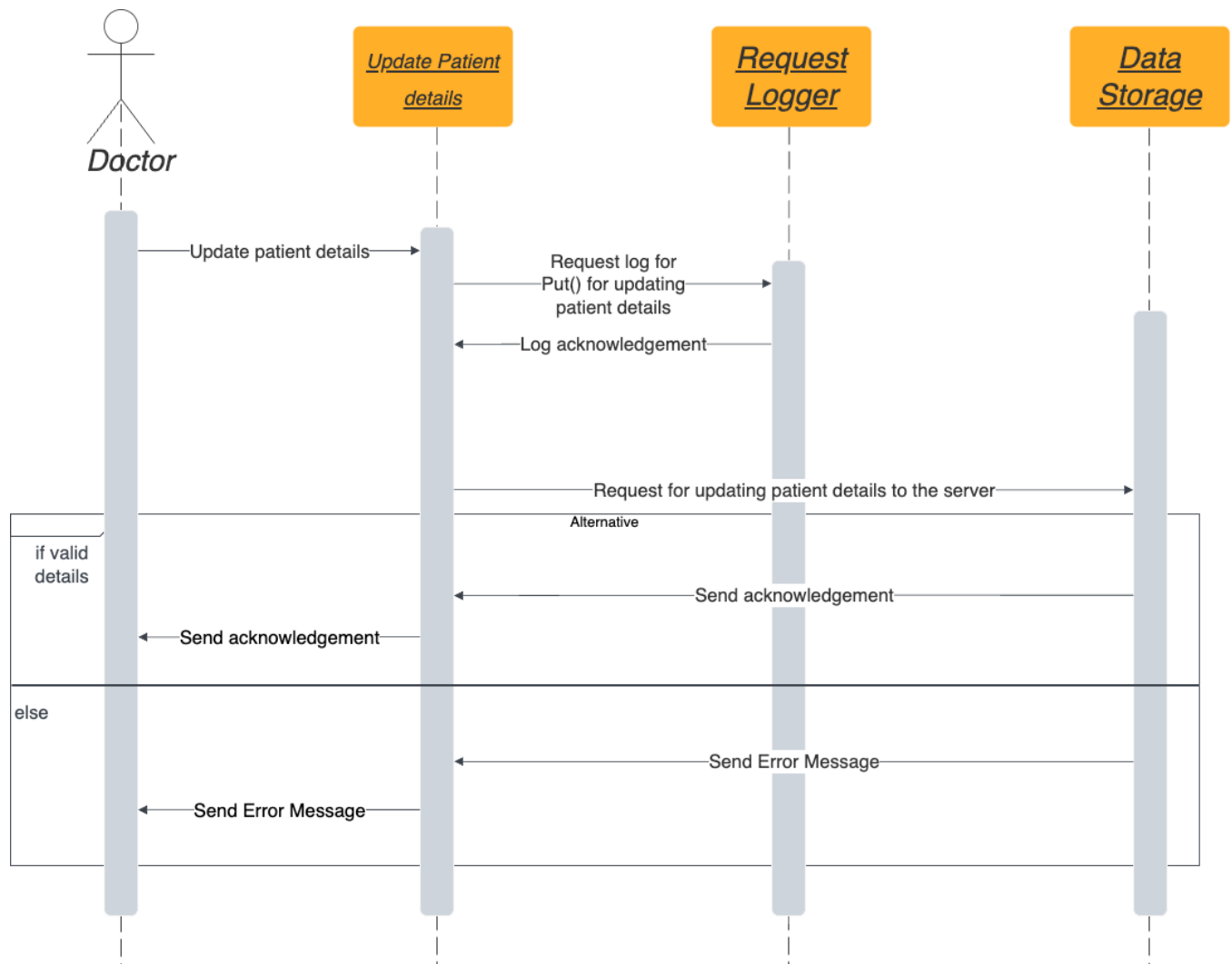
h) Doctor – view all patients



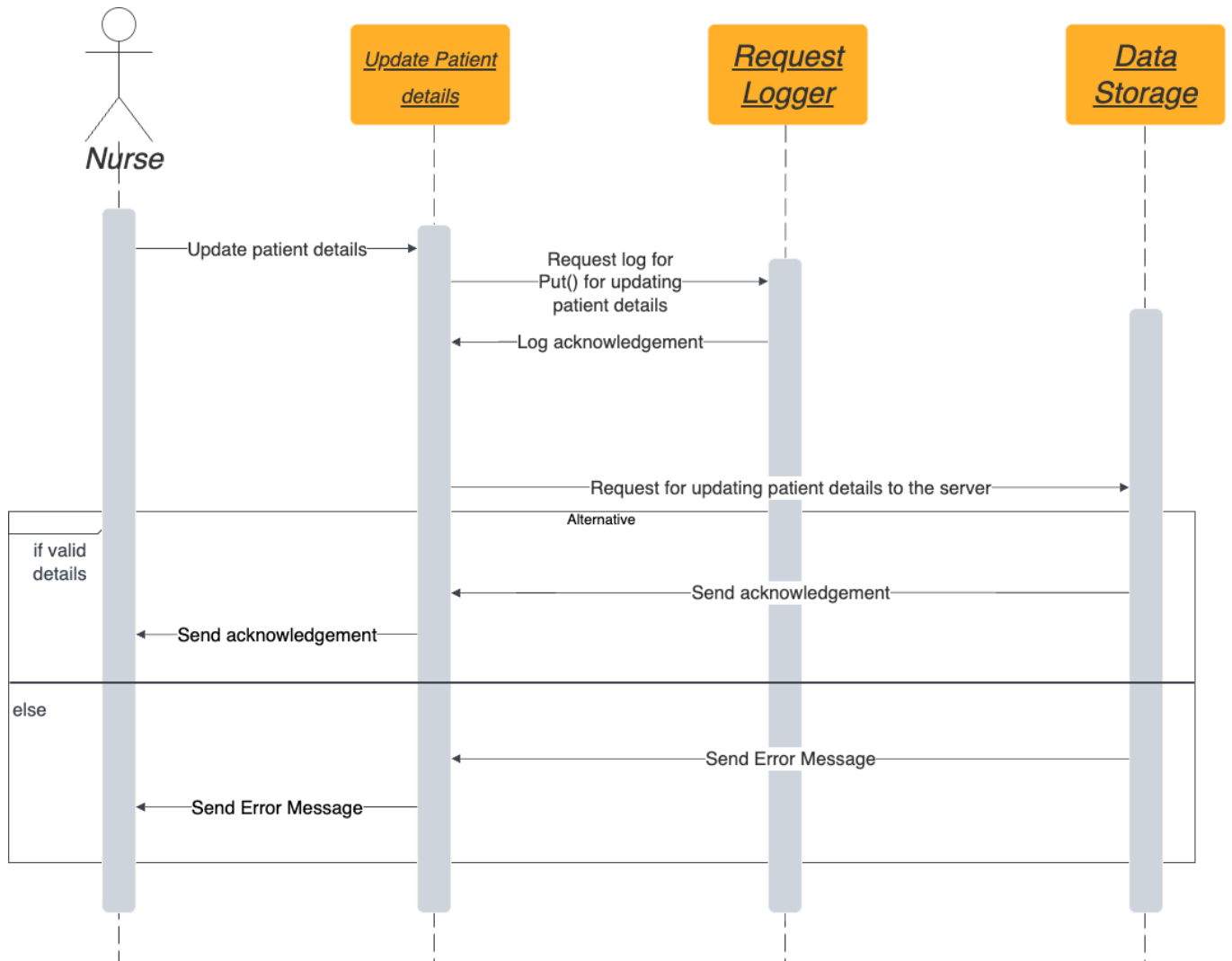
i) Nurse – view all patients



j) Doctor – update patient information

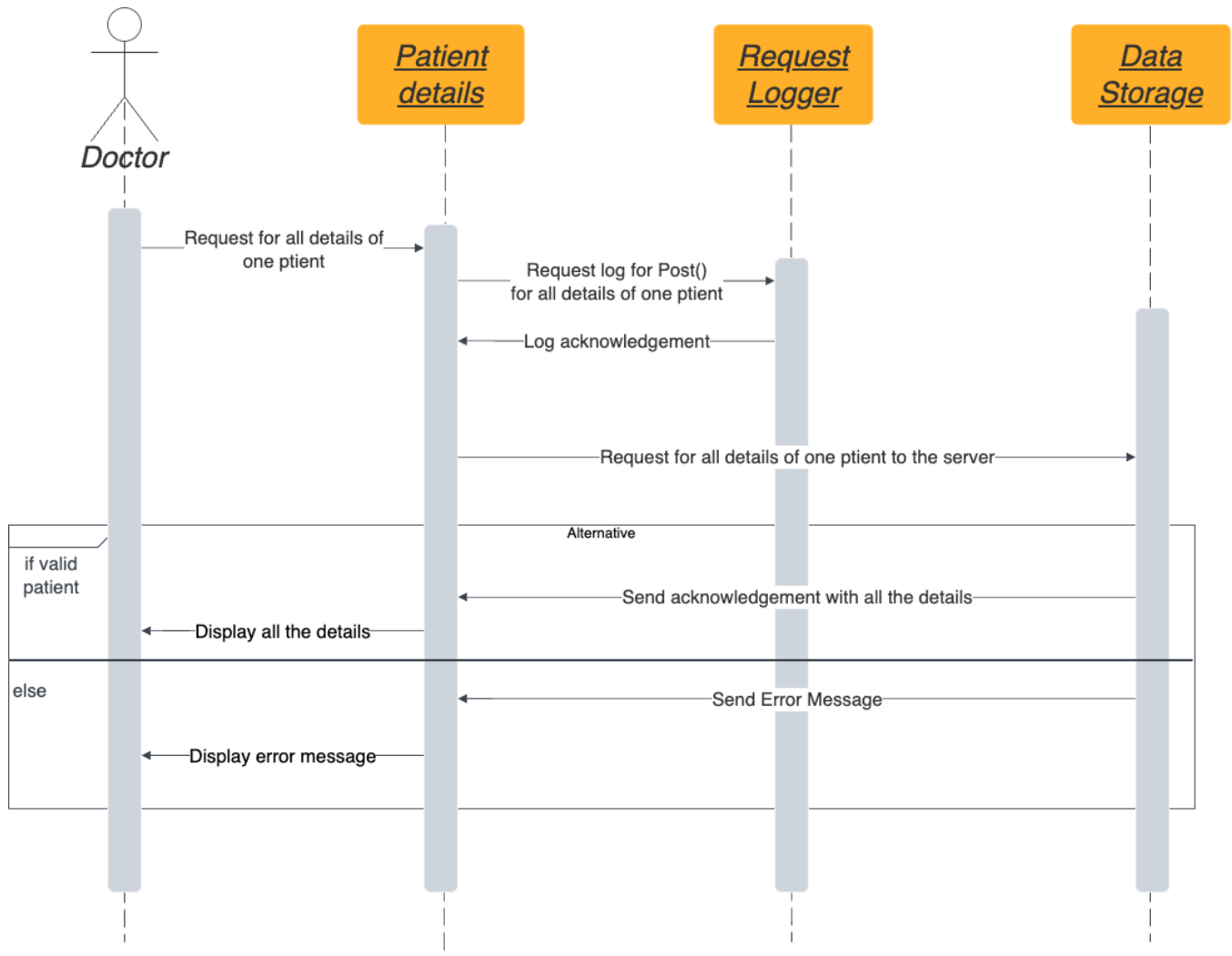


k) Nurse – update patients information

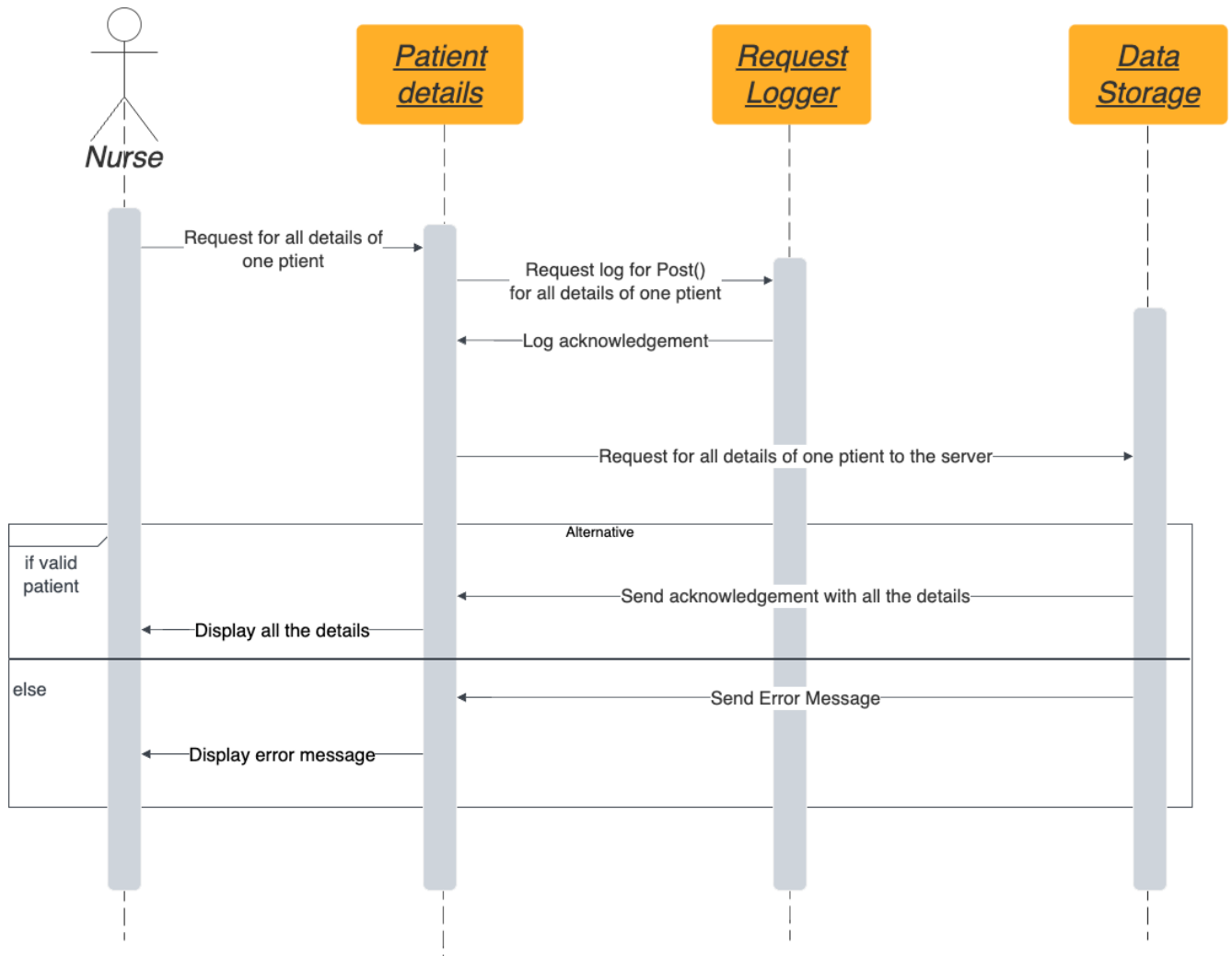




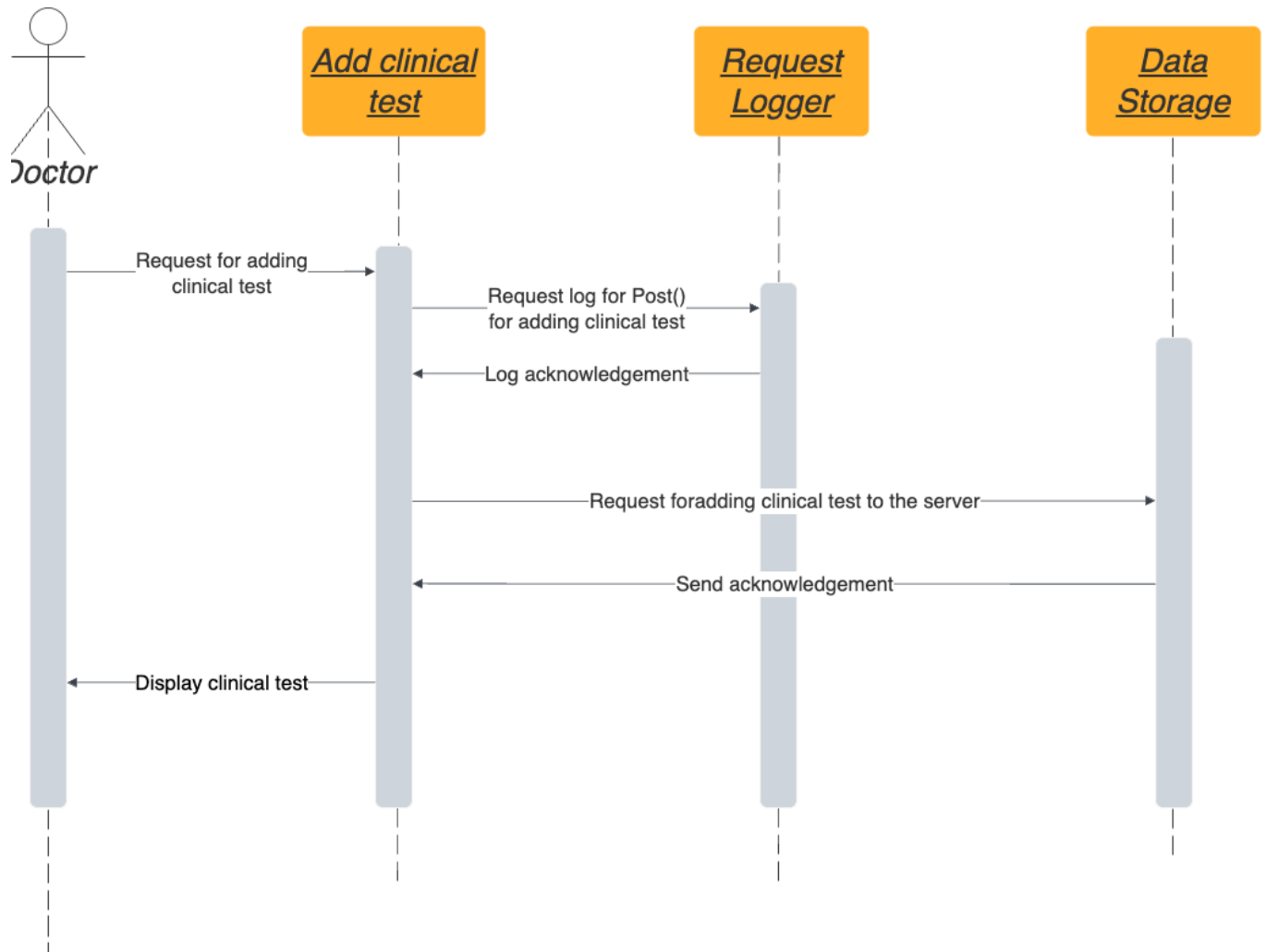
l) Doctor – patient information



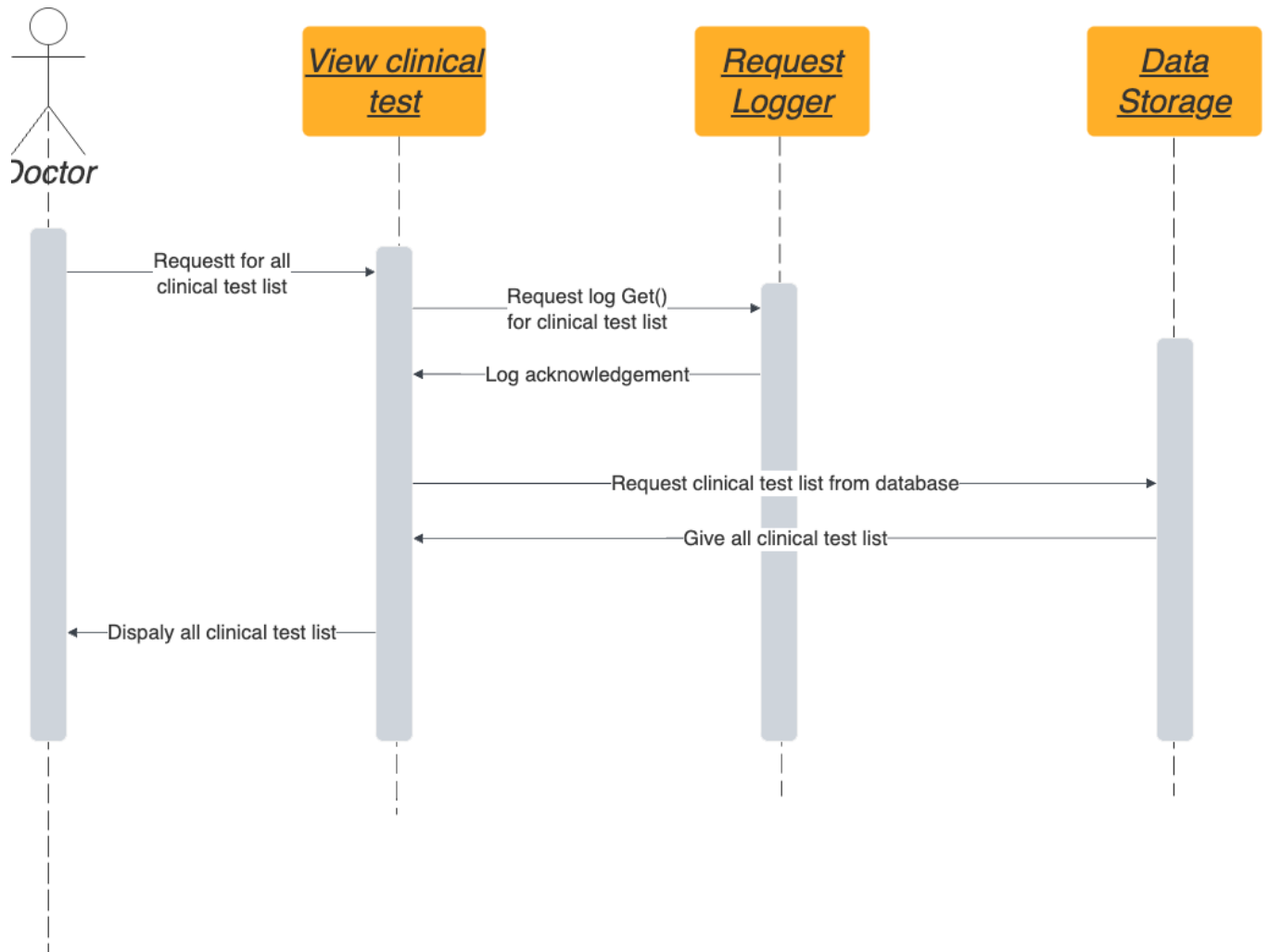
m) Nurse – patient information



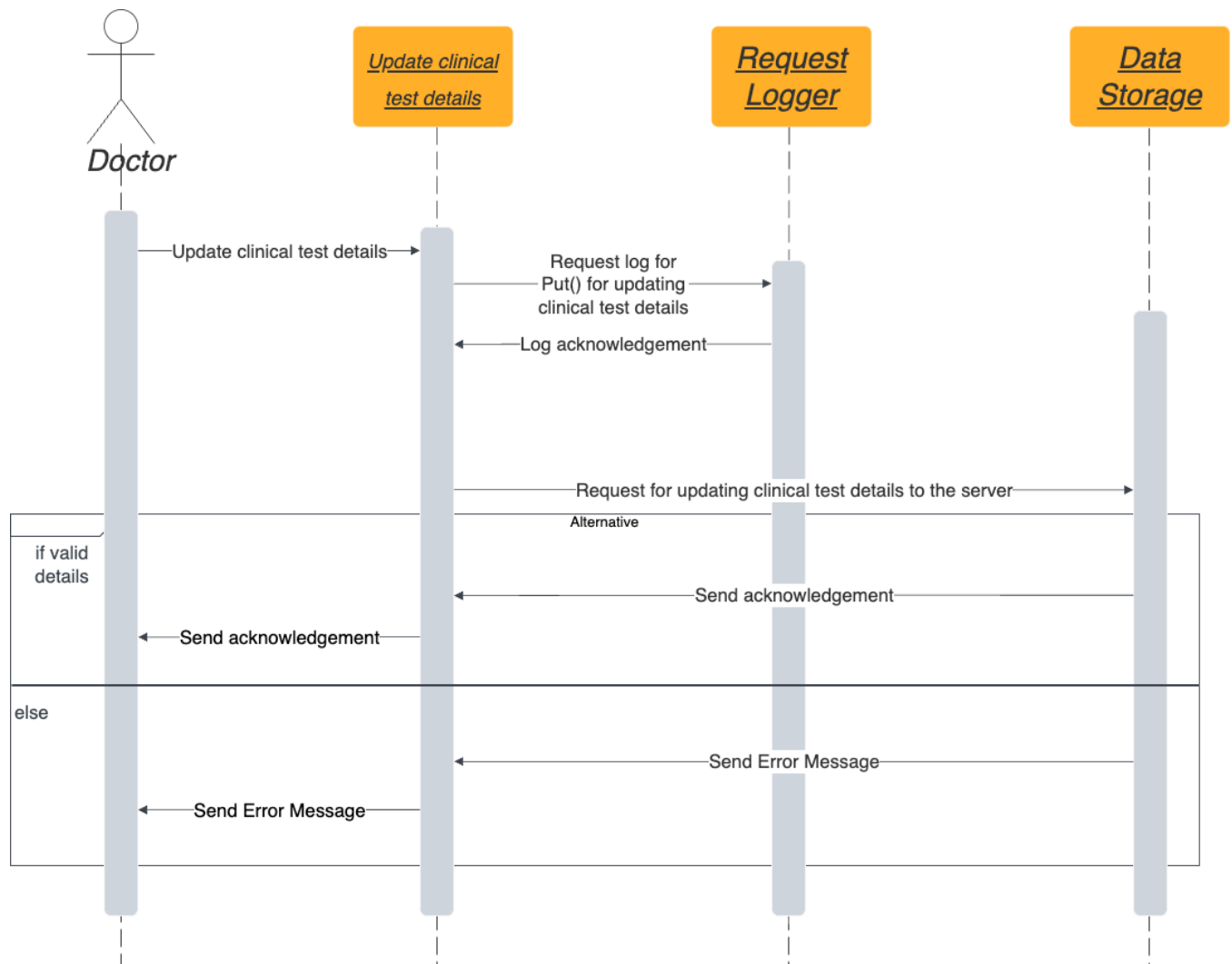
n) Doctor – add clinical tests



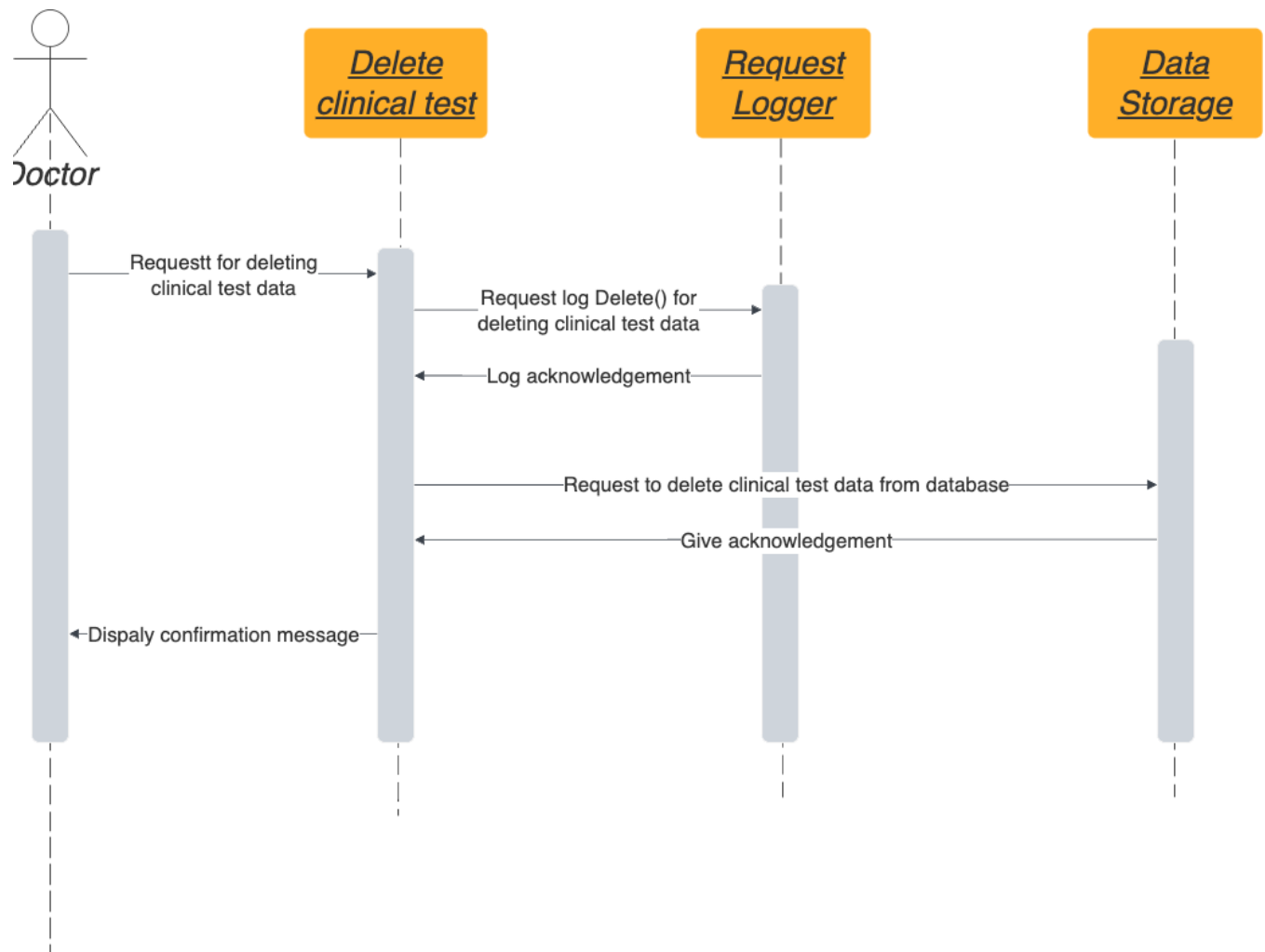
o) Doctor – view clinical tests



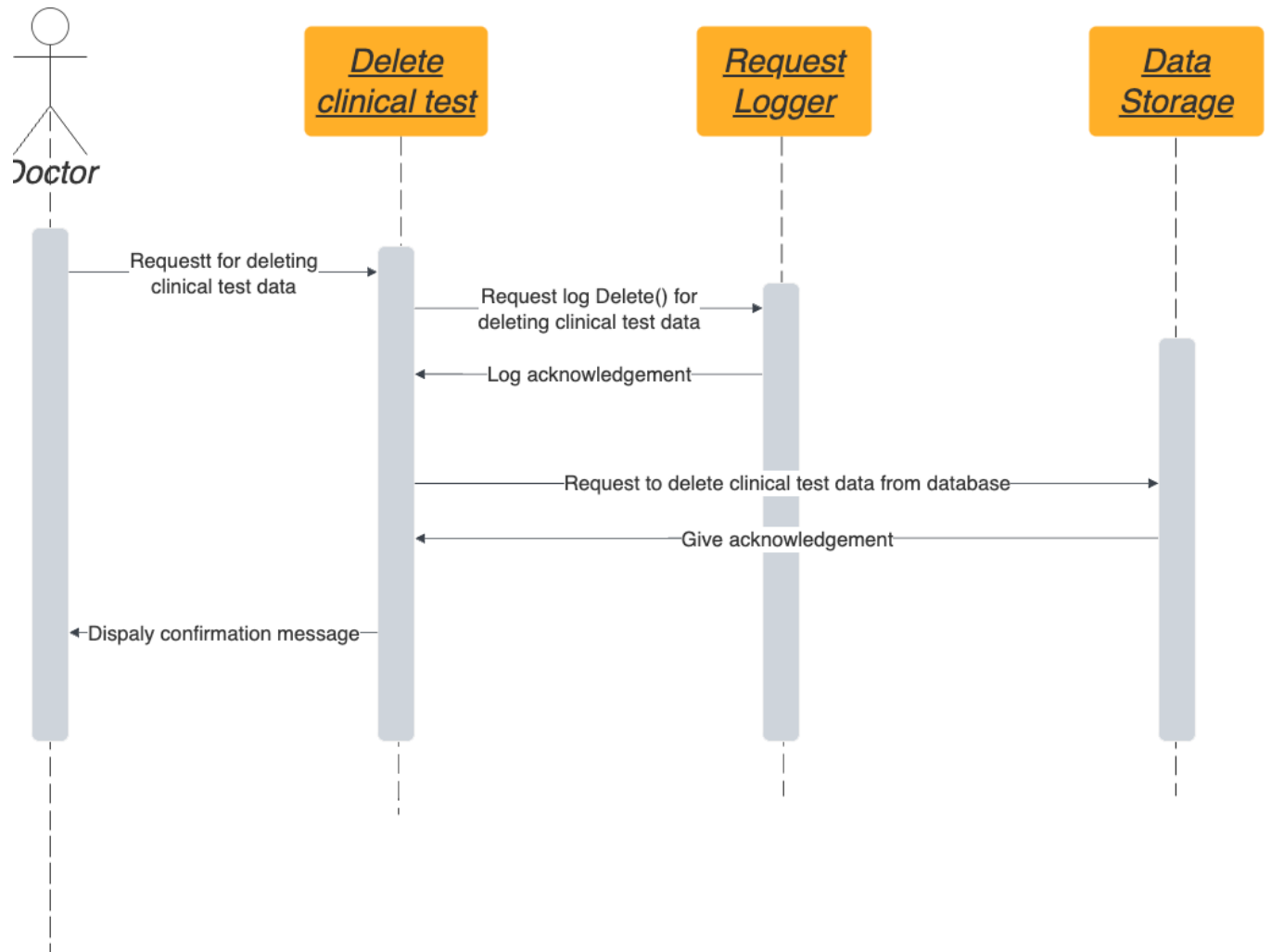
p) Doctor – update clinical tests



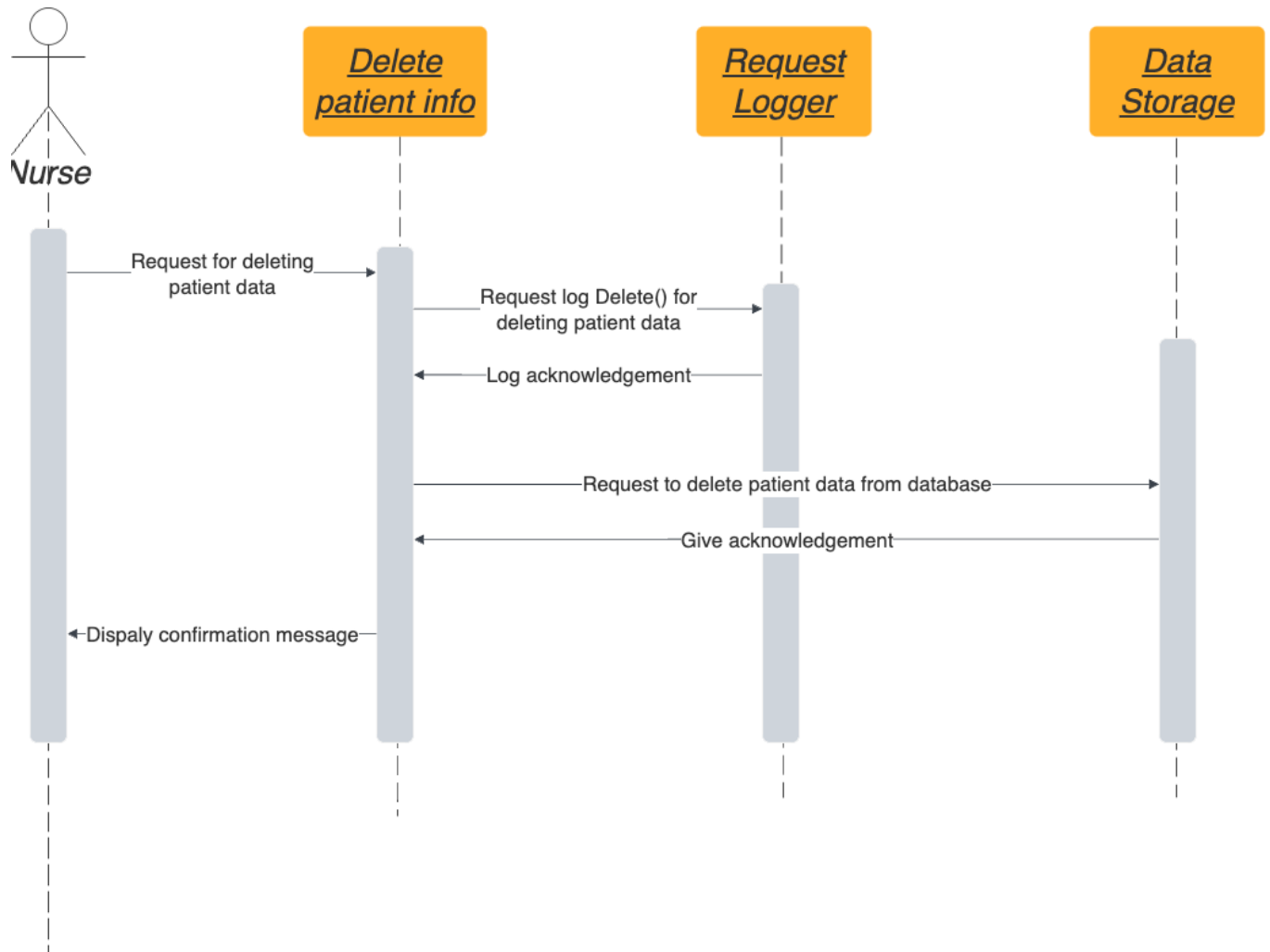
q) Doctor – delete clinical tests



r) Doctor – delete patients information

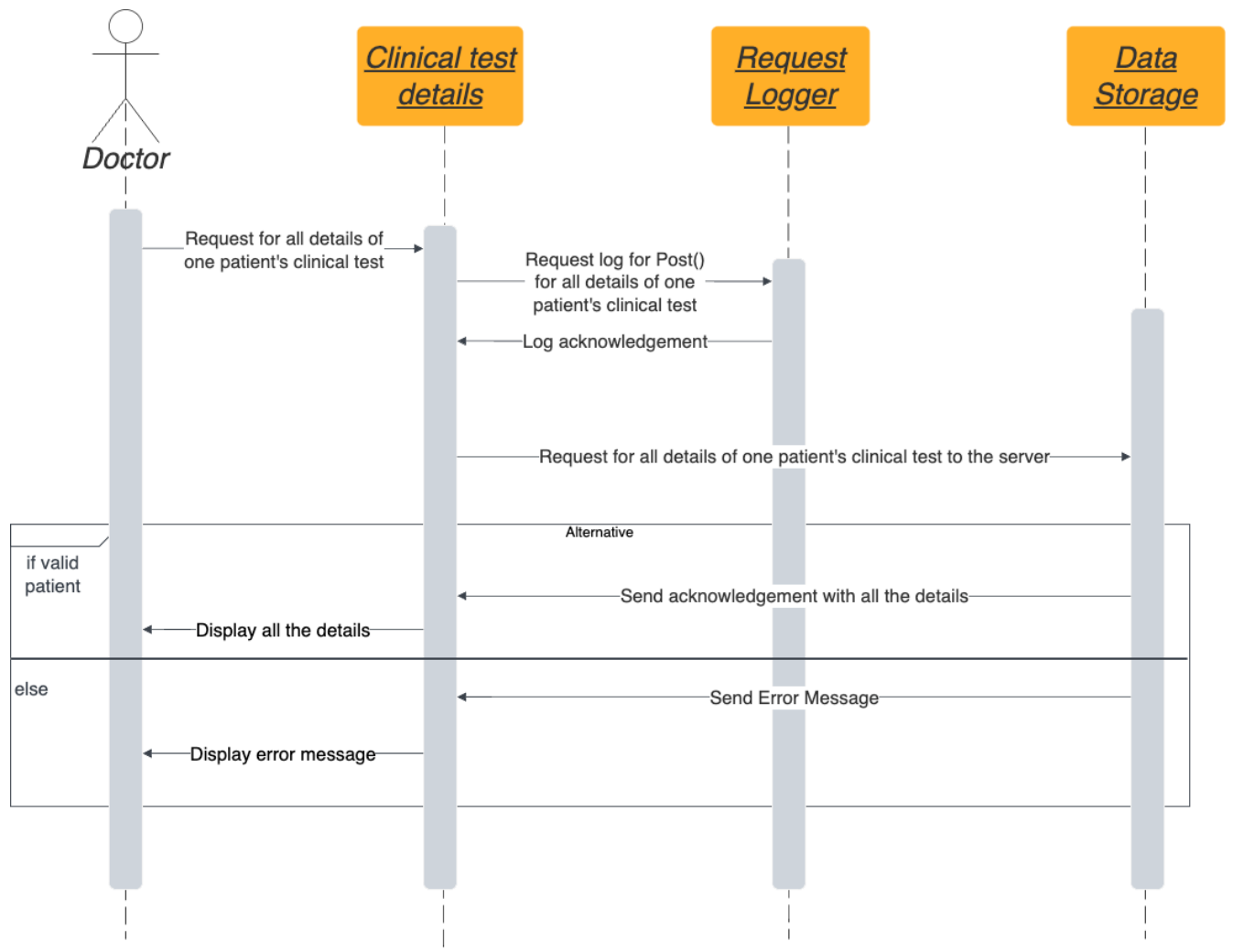


s) Nurse – delete patients information

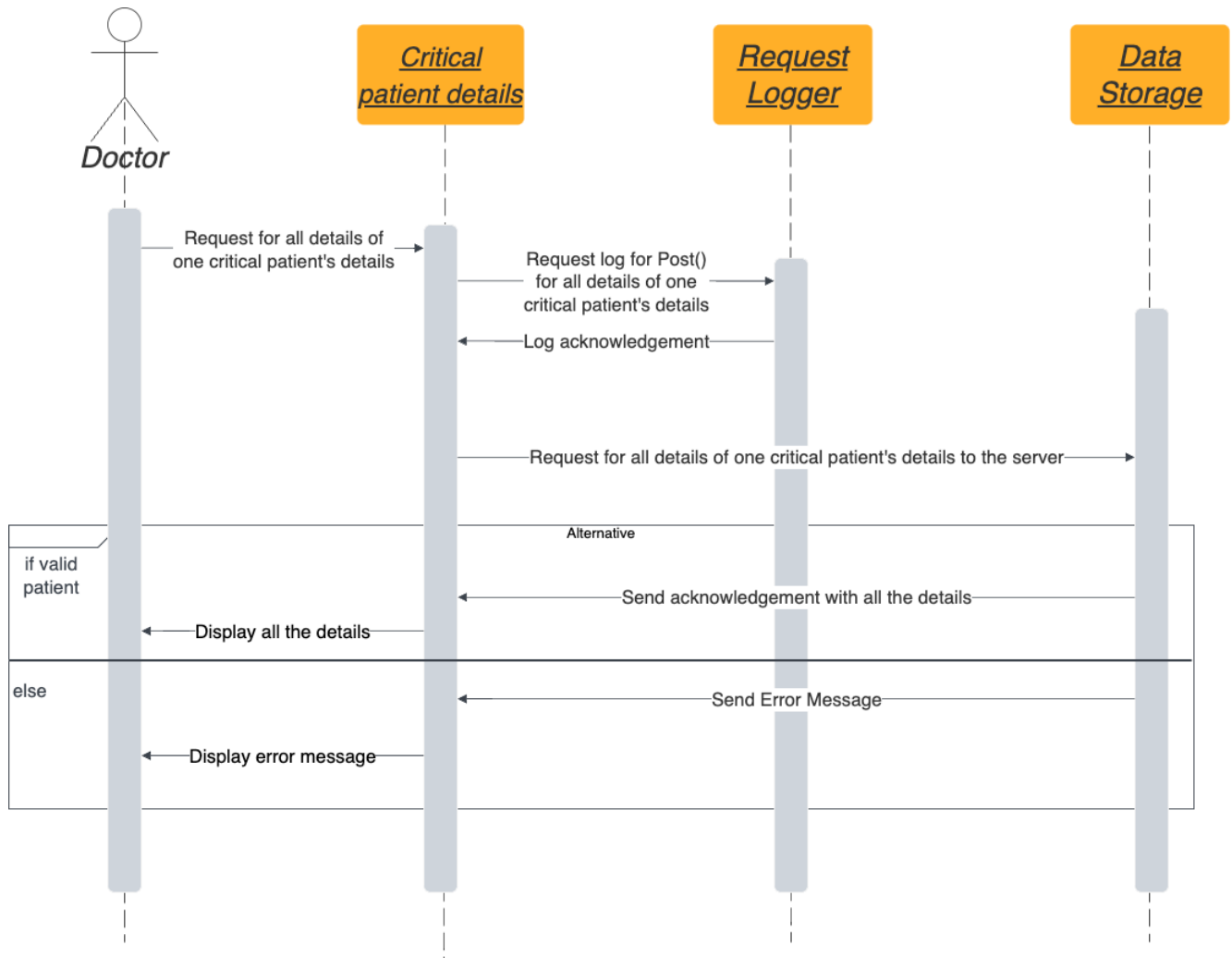




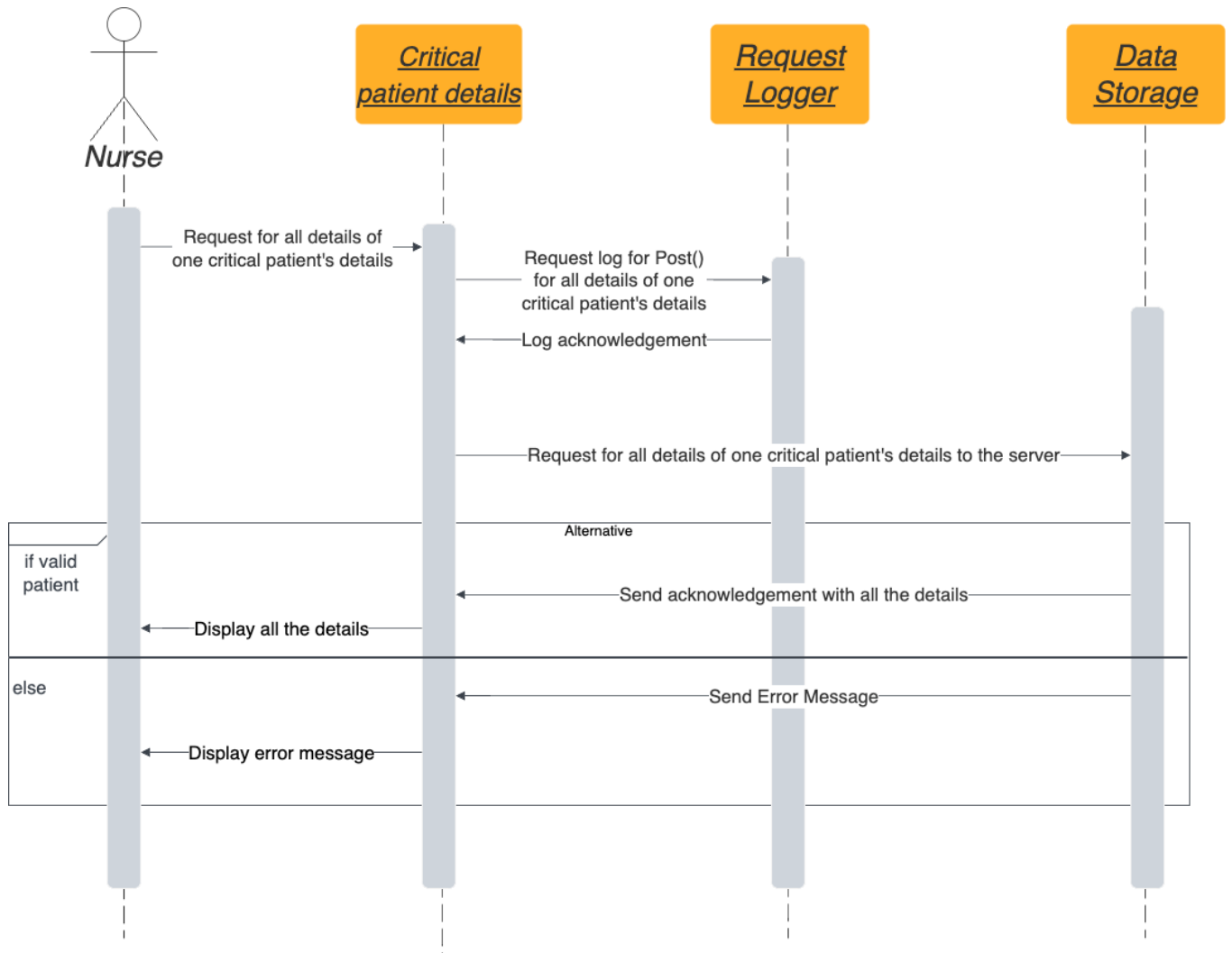
t) Doctor – view specific patients clinical tests



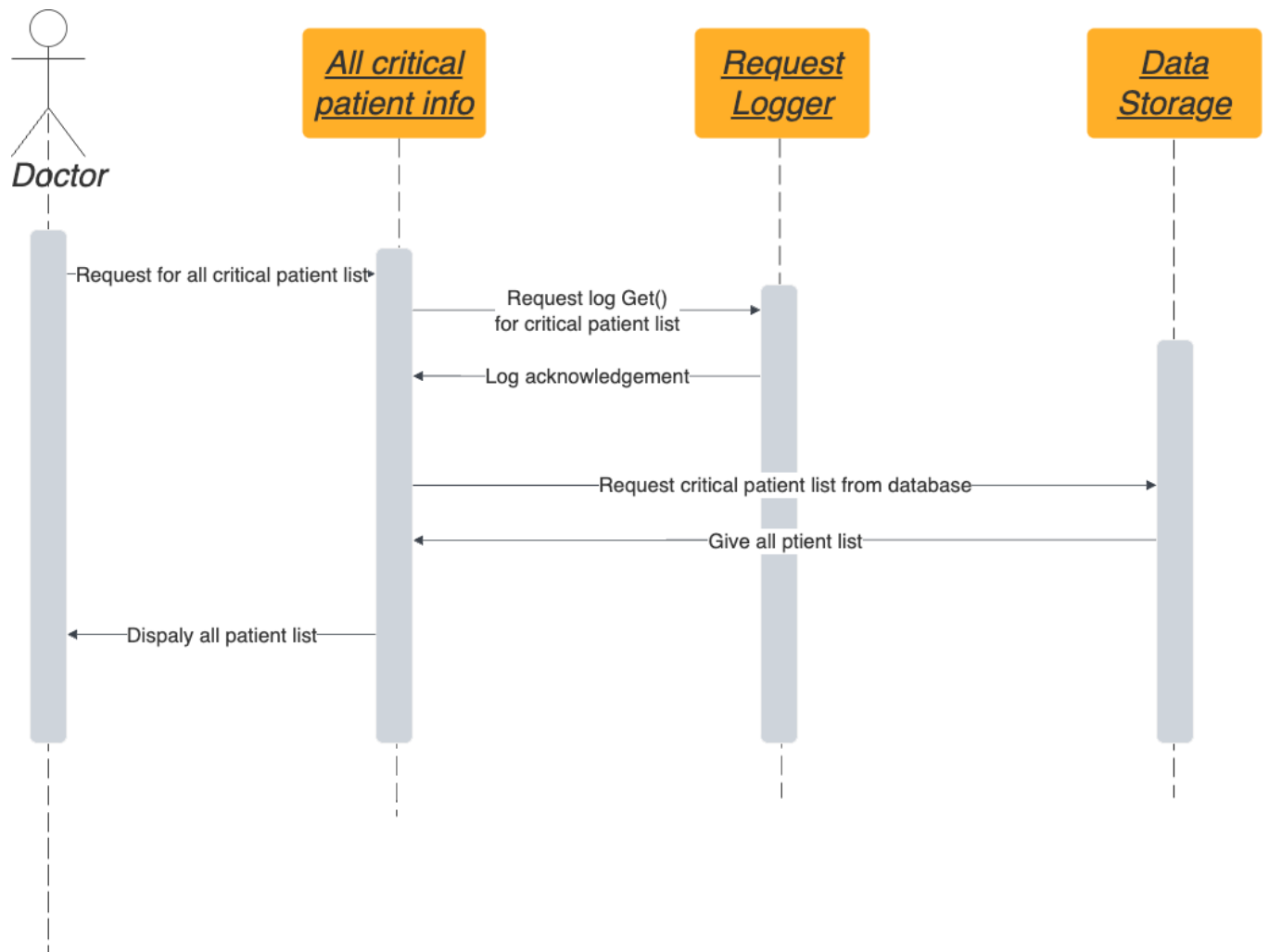
u) Doctor – view specific critical patients list



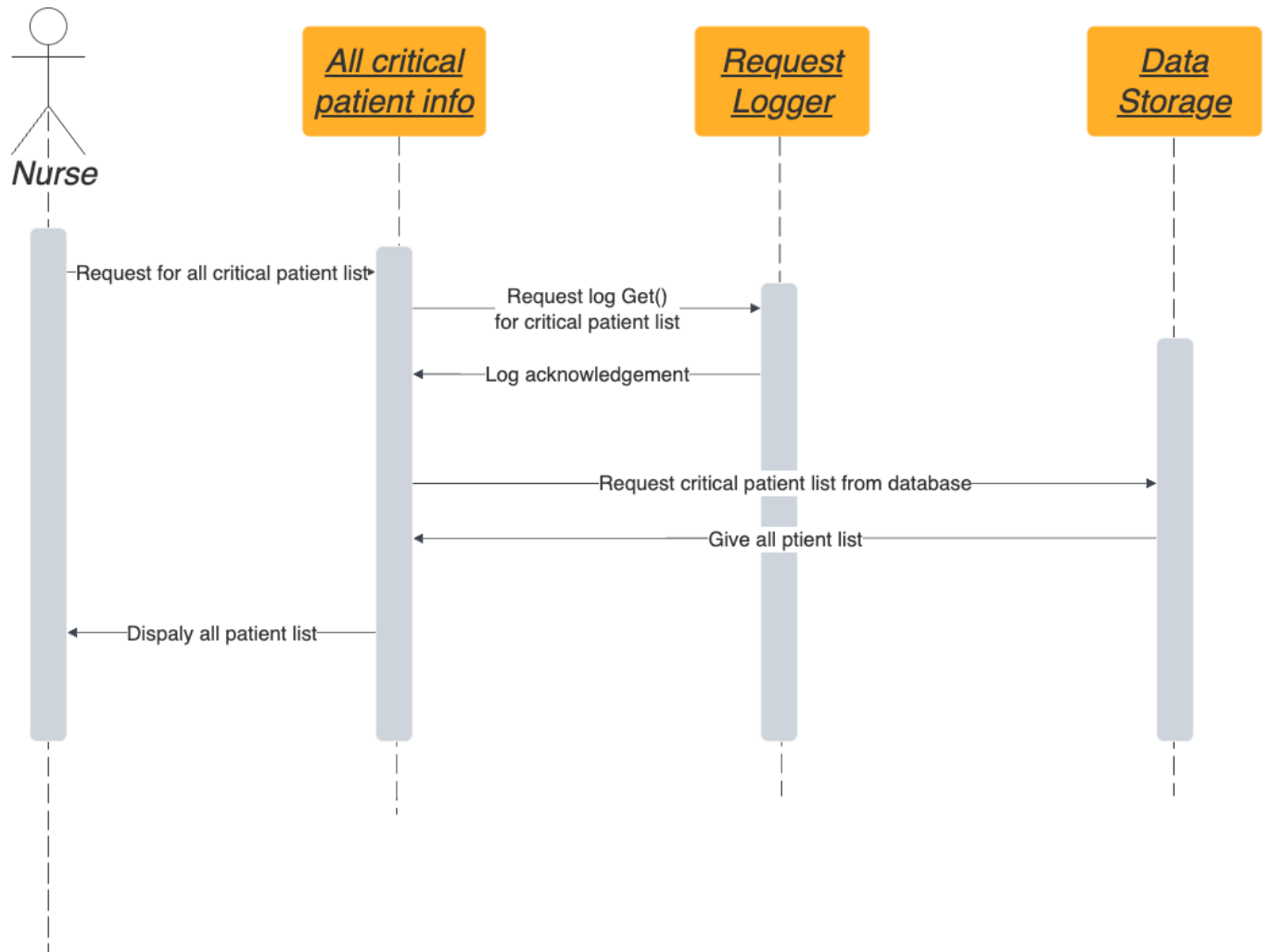
v) Nurse – view specific critical patients list



w) Doctor – view all critical patients



x) Nurse – view all critical patients



## 5. Endpoints of API

- [http://127.0.0.1:5000/register\\_patient](http://127.0.0.1:5000/register_patient)
- <http://127.0.0.1:5000/login>
- [http://127.0.0.1:5000/edit\\_profile/{:id}](http://127.0.0.1:5000/edit_profile/{:id})
- [http://127.0.0.1:5000/add\\_patient](http://127.0.0.1:5000/add_patient)
- [http://127.0.0.1:5000/view\\_patient/{:id}](http://127.0.0.1:5000/view_patient/{:id})
- [http://127.0.0.1:5000/edit\\_patient/{:id}](http://127.0.0.1:5000/edit_patient/{:id})
- [http://127.0.0.1:5000/delete\\_patient/{:id}](http://127.0.0.1:5000/delete_patient/{:id})
- [http://127.0.0.1:5000/patient\\_list](http://127.0.0.1:5000/patient_list)
- [http://127.0.0.1:5000/add\\_test/{:id}](http://127.0.0.1:5000/add_test/{:id})
- [http://127.0.0.1:5000/view\\_test/{:id}/{:test\\_id}](http://127.0.0.1:5000/view_test/{:id}/{:test_id})
- [http://127.0.0.1:5000/edit\\_test/{:id}/{:test\\_id}](http://127.0.0.1:5000/edit_test/{:id}/{:test_id})
- [http://127.0.0.1:5000/delete\\_test/{:id}/{:test\\_id}](http://127.0.0.1:5000/delete_test/{:id}/{:test_id})
- [http://127.0.0.1:5000/critical\\_patient](http://127.0.0.1:5000/critical_patient)
- [http://127.0.0.1:5000/critical\\_patient/{:id}](http://127.0.0.1:5000/critical_patient/{:id})