

# ECHO GROUP

## 1 Use case 1

ID: CPM:US1.001

Title: View vitals measurements

Actors: Actors: Doctor, nurse, social and health workers (only for home records), Caregivers (only for home records), Patient (only for home records)

Precondition:

Main scenario:

1. The user selects the 'View vitals measurements' tab for the given Patient.
2. The system searches for the measurements and displays the result list.
3. The actor browses the results.

Alternative flow no. 01:

1. If at step 3 of the main scenario, the actor selects the sub-option 'Add a measurement,' include the use case CPM:US1.002– Add a measurement.

Alternative flow no. 02:

1. If at step 3 of the main scenario, the actor selects the 'display trend,' include the use case CPM:US1.003– Display the graph of the measurements.

Exceptions:

### 1.1 ChatGPT last Improved Version

ID: CPM:US1.001

Title: View vitals measurements

Actors: Doctor, nurse, healthcare professionals (only for home records), caregivers (only for home records), patient (only for home records)

**Preconditions:**

- The patient must be registered in the system
- There must be existing measurements for the patient.

Main scenario:

1. The actor selects the 'View vitals measurements' tab for the given patient.
2. The system prompts the actor to select a date range or measurement type.
3. The actor selects a valid date range or measurement type.
4. The system retrieves the measurements from the database.
5. The system displays the measurements to the actor.
6. The actor navigates the results.

Alternative flow 01:

1. At step 5 of the main scenario, the actor selects the sub-option 'Add a measurement.'
2. The system transitions to the use case CPM:US1.002 – Add a measurement.

Alternative flow 02:

1. At step 5 of the main scenario, the actor selects the 'display trend.'
2. The system transitions to the use case CPM:US1.003 – Display the graph of the measurements.

**Error handling:**

# ECHO GROUP

- If the patient is not registered in the system, the system displays an error message and prompts the actor to register the patient.
- If there are no measurements found for the patient, the system displays a message indicating that there are no measurements available for the selected date range or measurement type.
- If the input date range or measurement type is invalid, the system displays an error message and prompts the actor to enter a valid date range or select a valid measurement type.

**Postconditions:** The system displays the measurements to the actor, and the actor can view or manipulate the data as needed. step 5 of the main scenario.

## 1.2 First ChatGPT suggestions

Completeness:

- Actors: The actors are reported, but it could be improved by specifying the specific roles of social and health workers and caregivers (e.g., what is their involvement in the process).
- Preconditions: There are no preconditions reported, which could be improved by specifying if the patient needs to be registered in the system and if there needs to be existing measurements for the patient.
- Postconditions: There are no postconditions reported, which could be improved by specifying the state of the system after the use case has been executed (e.g., does the system return to the previous screen, or does it remain on the same screen?).
- Basic Flow: The basic flow is not complete as it lacks details about how the system searches for the measurements (e.g., is there a specific date range? Are there different types of measurements available?). Input data validation is not reported. To improve atomicity, step 3 should be split into two steps: "The system retrieves the measurements from the database" and "The system displays the measurements to the actor."
- Alternative Flows: The alternative flows are complete, but it could be improved by specifying how the system transitions to the alternative flow (e.g., is there a button to add a measurement or display trend?).
- Error Handling: There is no error handling reported, which could be improved by specifying potential errors that may occur (e.g., system error, no measurements found for the patient).

Correctness:

- Actors: The actors are correctly identified.
- Preconditions: The preconditions are not reported, so it is unclear if they accurately identify the success criteria for the main scenario.
- Postconditions: The postconditions are not reported, so it is unclear if they accurately describe the state of the system after the use case has been executed.
- Basic Flow: The basic flow is correctly ordered, but it lacks details about how the system searches for the measurements. To improve atomicity, step 3 should be split into two steps. Input data validation is not reported, so it is unclear if the system verifies the input data for correctness.
- Alternative Flows: The alternative flows are correctly ordered and atomic.
- Error Handling: There is no error handling reported, so it is unclear if the use case covers all potential errors that may occur.

Clarity:

- The use case is easily understandable for all stakeholders, including developers and non-developers.
- There is no extraneous information or jargon that could impede understanding.
- The terminology used in the use case is consistent. However, the use case refers to the user as "the user" and "the actor" interchangeably. It should consistently use "actor" as the actor throughout the use case.

# ECHO GROUP

- The steps in the use case are arranged in a logical sequence.

## 1.3 Second ChatGPT suggestions

### Completeness:

- Actors: The actors are reported and seem to cover all the possible users of the system.
- Preconditions: The preconditions are reported and seem complete.
- Postconditions: The postconditions are reported and seem complete.
- Basic Flow: The basic flow is complete. However, it lacks input data validation, which should be added.
- Alternative Flows: The alternative flows are reported and seem complete.
- Error Handling: The error handling is reported and seems complete.

### Correctness:

- Actors: The actors are correctly identified.
- Preconditions: The preconditions are accurate and correctly identify the success criteria for the main scenario of the use case.
- Postconditions: The postconditions accurately describe the state of the system after the use case has been executed.
- Basic Flow: The basic flow is correctly ordered, but it is not atomic. To improve atomicity, the following changes should be made:
  1. Step 2: split into two steps: a) the system prompts the actor to select a date range or measurement type; b) the actor selects a valid date range or measurement type.
  2. Step 4: split into two steps: a) the system retrieves the measurements from the database; b) the system displays the measurements to the actor.
- Alternative Flows: The alternative flows are correctly ordered and atomic.
- Error Handling: The error handling accurately covers all potential errors that may occur.

### Clarity:

- The use case is easily understandable for all stakeholders, including developers and non-developers. There is no extraneous information or jargon that could impede understanding.
- The use case is consistent in terms of logical sequence and terminology. However, the term "browses" could be replaced by "navigates" to avoid confusion.

## 2 Use Case 2

ID: CPM:US1.002

Title: Add a measurement

Actors: Doctor, nurse, patient, caregiver, social health operator

Precondition:

Main scenario:

1. The user clicks "New measurement".
2. The system displays a form in which it is possible to select the desired measuring instrument.
3. The user selects the desired measuring instrument.
4. The system searches and displays the vital parameters associated with that measuring instrument.
5. The user fills in the fields.
6. The user presses the "Save" button.
7. The system saves the changes for the new measurement to the list of measurements for the selected vital parameter.
8. The system displays the message "The measurement has been saved correctly".

# ECHO GROUP

9. The system displays the refreshed list of measurements with the new measurement.

Alternative flow no. 01:

1. At step 4 of the main scenario, the user presses the button "Cancel".
2. The system closes the form.

Exceptions:

## 2.1 ChatGPT improved Version

ID: CPM:US1.002

Title: Add a measurement

Actors: Doctor, nurse, patient, caregiver, social health operator

**Preconditions:**

- The user is authenticated and authorized to perform this action.
- The system has a list of measuring instruments and their corresponding vital parameters

**Postconditions:** The system displays the new measurement in the patient's medical record.

Main scenario:

1. The user clicks the "New Measurement" button.
2. The system displays a form with a list of measuring instruments.
3. The user selects the desired measuring instrument from the list.
4. The system searches the vital parameters associated with the selected measuring instrument.
5. The system displays the vital parameters associated with the selected measuring instrument.
6. The user selects a vital parameter field from the list.
7. The user enters numerical values for the selected vital parameter field.
8. The user clicks the "Save" button.
9. The system validates the user input.
10. The system saves the new measurement to the list of measurements for the selected vital parameter.
11. The system displays the message "The measurement has been saved correctly".
12. The system displays the refreshed list of measurements with the new measurement.

Alternative flow no. 01:

1. At step 4 of the main scenario, the user clicks the "Cancel" button.
2. The system closes the form.

**Error handling:**

- If the system cannot save the new measurement correctly, it displays the error message "Unable to save the measurement. Please try again".
- If the user enters invalid data, the system displays the error message "Invalid input. Please enter a valid value".

## 2.2 First ChatGPT suggestions

Completeness:

# ECHO GROUP

## Actors:

- The actors are reported, and they are doctor, nurse, patient, caregiver, and social health operator.

## Preconditions:

- There are no preconditions reported. It would be useful to add a precondition that the user needs to be authenticated and authorized to perform this action.

## Postconditions:

- The postconditions are missing. It would be useful to include what the system should do after the use case has been executed. For example, the system should display the new measurement in the patient's medical record.

## Basic Flow:

- The basic flow is complete, but it would be useful to add input data validation to step 5 to ensure that the fields are filled in correctly. For example, it could be useful to specify that the system should only accept numerical values for vital parameters.
- To improve atomicity, step 5 could be divided into two steps: 'The user fills in the fields with the vital parameters' and 'The user verifies that the values entered are correct before proceeding to the next step.'

## Alternative Flows:

- The only alternative flow reported is complete and states that the user can cancel the action at step 4.

## Error Handling:

- The error handling is missing. It would be useful to add some error handling to the use case, for example, to handle cases where the system cannot save the new measurement correctly or the user enters invalid data.

## Correctness:

# ECHO GROUP

## Actors:

- The actors are correctly identified.

## Preconditions:

- The preconditions are missing. It would be useful to add a precondition that the user needs to be authenticated and authorized to perform this action.

## Postconditions:

- The postconditions are missing. It would be useful to include what the system should do after the use case has been executed. For example, the system should display the new measurement in the patient's medical record.

## Basic Flow:

- The basic flow is correctly ordered, but it can be improved to improve atomicity. To improve atomicity, step 5 could be divided into two steps: 'The user fills in the fields with the vital parameters' and 'The user verifies that the values entered are correct before proceeding to the next step.'

## Alternative Flows:

- The only alternative flow reported is correctly ordered.

## Error Handling:

- The error handling is missing. It would be useful to add some error handling to the use case, for example, to handle cases where the system cannot save the new measurement correctly or the user enters invalid data.

## Clarity:

- The use case is easily understandable for all stakeholders, including developers and non-developers. However, it would be useful to remove any extraneous information that does not relate

# ECHO GROUP

to the use case. For example, it is not necessary to state that the user clicks "New measurement." Instead, it could be stated that the user starts the process of adding a new measurement.

- The use case is consistent in terms of logical sequence and terminology consistency. However, it would be useful to use consistent terminology throughout the use case. For example, the use of the term "user" and "patient" could be made consistent.

## 2.3 Second ChatGPT suggestions

Completeness:

- Actors: Actors are reported, and they include Doctor, nurse, patient, caregiver, social health operator.
- Preconditions: The precondition is reported, and it states that the user is authenticated and authorized to perform this action.
- Postconditions: Postconditions are reported, and they state that the system displays the new measurement in the patient's medical record.
- Basic Flow: The basic flow appears to be complete. However, it could be improved by adding a step for input data validation.
- Alternative Flows: The alternative flow is present, and it indicates that the user can cancel the process at step 4.
- Error Handling: The error handling is reported, and it includes displaying an error message when the system cannot save the new measurement correctly, and when the user enters invalid data.

Suggestions for improvement:

- Basic Flow: Add a step for input data validation to ensure that the user enters valid data before saving the new measurement.
- Alternative Flows: None.
- Error Handling: Add specific error messages for each type of error that may occur, so that users can quickly identify and fix the issue.

Correctness:

- Actors: The actors are correctly identified.
- Preconditions: The precondition is accurate and correctly identifies the success criteria for the main scenario of the use case.

# ECHO GROUP

- Postconditions: The postconditions accurately describe the state of the system after the use case has been executed.
- Basic Flow: The basic flow is correctly ordered and atomic. However, step 6 could be split into two steps to improve atomicity. For example, "The user fills in the vital parameter fields with numerical values" could be split into "The user selects the vital parameter field" and "The user enters numerical values for the selected vital parameter field".
- Alternative Flows: The alternative flow is correctly ordered and atomic.
- Error Handling: The error handling covers potential errors that may occur.

Suggestions for improvement:

- Basic Flow: Split step 6 into two steps to make the use case more atomic.
- Alternative Flows: None.
- Error Handling: None.

Clarity:

- The use case is easily understandable for all stakeholders, including developers and non-developers. There is no extraneous information or jargon that could impede understanding.
- The use case is consistent in terms of terminology and sequence of events.

## 3 Use case 3

ID: CPM:US1.003

Title: Display the graph of the measurements for a vital parameter

Actors: Doctor, nurse, Social health operator, caregiver, patient

Preconditions:

Main scenario:

1. The user selects "Display measurement graph for vital" for the given parameter.
2. The system searches the measurements for the vital parameter and displays the related graph, having the measures on the ordinates, and the date-time on the abscissas.

### 3.1 ChatGPT last improved version

ID: CPM:US1.003

Title: Display the graph of the measurements for a vital parameter

Actors: Doctor, nurse, Social health operator, caregiver, patient (with permission to access)



# ECHO GROUP

## Preconditions:

- The user has the permission to access this use case.
- The system has the necessary data to display the graph.

## Postconditions:

- The graph of the measurements for the selected vital parameter is displayed.
- The user can interact with the graph by zooming in/out, selecting a specific date range, or other interactions.

## Basic Flow:

1. The user selects "Display measurement graph for vital" for the given parameter.
2. The system validates the input data:
  - a. Check if the selected vital parameter exists.
  - b. Check if the user has selected a valid date range.
3. The system checks if the user has permission to access the use case.
4. The system searches the measurements for the vital parameter.
5. The system displays the related graph, having the measures on the ordinates, and the date-time on the abscissas.
6. The user interacts with the graph as needed.

## Alternative Flow no. 1:

- At step 4, the user closes the Form.
- The system returns to the previous screen.

## Error Handling:

- If the user does not have the permission to access this use case, the system displays an error message and terminates the use case.
- If the system does not have the necessary data to display the graph, the system displays an error message and terminates the use case.
- If the input data is invalid, the system displays an error message and prompts the user to enter valid data.

## 3.2 First ChatGPT suggestions

### Completeness:

- Actors: The actors are listed, but it is not clear whether all of them have the permission to access this use case. It would be helpful to specify which actors have the permission to access this use case.
- Preconditions: The preconditions are missing. It would be helpful to include the preconditions to ensure that the user has the necessary permissions to access this use case and that the system has the necessary data to display the graph.
- Postconditions: The postconditions are missing. It would be helpful to include the expected outcome after the user has completed the use case.
- Basic Flow: The basic flow is simple and complete. However, there is no mention of input data validation in the basic flow. It would be helpful to include a step to validate the input data before proceeding. In addition, it would be helpful to add a new step to specify what the user does after the graph is shown.

# ECHO GROUP

- Alternative Flows: There are no alternative flows in this use case. However, it could be appropriate given its simplicity. A potential alternative flow could be that the user decides to return to the homepage at any point in the basic flow.
- Error Handling: The error handling is missing. It would be helpful to identify the potential errors that may occur during the use case and provide steps to handle those errors.

## Correctness:

- Actors: The actors are correctly identified, but it is not clear which actors have the permission to access this use case. It would be helpful to specify which actors have the permission to access this use case.
- Preconditions: The preconditions are missing. It is important to identify the preconditions to ensure that the user has the necessary permissions to access this use case and that the system has the necessary data to display the graph.
- Postconditions: The postconditions accurately describe the expected outcome after the use case has been executed.
- Basic Flow: The basic flow is correctly ordered but not atomic. It would be helpful to split the step "The system searches the measurements for the vital parameter and displays the related graph, having the measures on the ordinates, and the date-time on the abscissas" into two steps: "The system searches the measurements for the vital parameter" and "The system displays the related graph, having the measures on the ordinates, and the date-time on the abscissas". This will ensure that the user and the system execute only one action in each step.
- Alternative Flows: There are no alternative flows in this use case, which could be appropriate given its simplicity.
- Error Handling: The error handling is missing. It is important to identify the potential errors that may occur during the use case and provide steps to handle those errors.

## Clarity:

- The use case is easily understandable for all stakeholders, including developers and non-developers.
- There is no extraneous information or jargon that could impede understanding.
- The terminology used in the use case is consistent.
- The steps in the use case are arranged in a logical sequence.

## 3.3 Second ChatGPT suggestions

### Completeness:

- Actors: The actors are reported.
- Preconditions: The preconditions are reported.
- Postconditions: The postconditions are reported.
- Basic Flow: The basic flow is complete. However, it does not report any input data validation.
- Alternative Flows: The alternative flows are complete.
- Error Handling: The error handling is reported and complete.

### Suggestions for improvement:

- Basic Flow: It is suggested to report the input data validation in step 2. For example, the system could check if the selected vital parameter exists and if the user has selected a valid date range.

### Correctness:

- Actors: The actors are correctly identified.

# ECHO GROUP

- Preconditions: The preconditions accurately identify the success criteria for the main scenario of the use case.
- Postconditions: The postconditions accurately describe the state of the system after the use case has been executed.
- Basic Flow: The basic flow is correctly ordered and atomic. However, step 2 could be split into two steps: validate the input data and check if the user has permission to access the use case.
- Alternative Flows: There are no alternative flows.
- Error Handling: The error handling accurately covers all potential errors that may occur.

Suggestions for improvement:

- Basic Flow: Splitting step 2 into two steps will improve atomicity and make it easier to understand the flow of events.

Clarity:

- The use case is easily understandable for all stakeholders, including developers and non-developers. There is no extraneous information or jargon that could impede understanding.
- The use case is consistent in terms of logical sequence and terminology consistency.

## 4 Use case 4

ID: CPM:US1.004

Title: Add an automatic measurement

Actors: Patient, Sensor Device, Mobile App

Preconditions:

Main scenario:

1. The user uses an instrument from the kit to measure a vital sign.
2. The Sensor Device sends the raw data to the Mobile.
3. The Mobile App requests the System to save the measurement data.
4. The system saves the measurement.
5. The system sends the Mobile App the envelope data of the operation.
6. The mobile App displays a message to the user to inform him (or her) that the data have been saved.

### 4.1 ChatGPT last improved version

ID: CPM:US1.004

Title: Add an automatic measurement

Actors: Patient, Sensor Device, Mobile App, System

**Preconditions:**

- The Patient has installed the Mobile App on their mobile device.
- The Patient has the kit with the instrument to measure the vital sign.
- The Sensor Device is available and connected to the Mobile App.

Main scenario:

1. The Patient selects the type of vital sign to measure using the instrument from the kit.
2. The Sensor Device starts measuring the vital sign and sends the raw data to the Mobile App.
3. The Mobile App receives the raw data from the Sensor Device and validates the measurement data.
4. The Mobile App sends a request to the System to save the measurement data.
5. The System saves the measurement data.

# ECHO GROUP

6. The System sends the Mobile App the envelope data of the operation.
7. The Mobile App displays a message to the Patient to inform them that the data have been saved.

**Alternative** flow no. 01:

- 2a. The user closes the app.
- 3a. The system does not save any data.

**Error Handling:**

- If the Sensor Device fails to send the raw data to the Mobile App, the Mobile App displays an error message to the Patient and requests them to repeat the measurement.
- If the Mobile App loses connectivity with the System, the Mobile App displays an error message to the Patient and requests them to repeat the measurement.
- If the raw data validation fails, the Mobile App displays an error message to the Patient and requests them to repeat the measurement.

**Postconditions:**

- The measurement data are saved in the System.
- The Mobile App informs the Patient that the data have been saved.

## 4.2 First ChatGPT suggestions

Completeness:

- Actors: The actors are reported (Patient, Sensor Device, Mobile App).
- Preconditions: The preconditions are missing. It is important to report if the patient has already installed the Mobile App and if the Sensor Device is available.
- Postconditions: The postconditions are missing. It would be helpful to include the expected outcome after the user has completed the use case.
- Basic Flow: The basic flow is incomplete, and it is missing the validation of the measurement data before saving. It is recommended to add a step to validate the measurement data before saving.
- Alternative Flows: There are no alternative flows. Specifying what happens if the user closes the Mobile App would be helpful.
- Error Handling: The error handling is missing. It is essential to consider potential errors that may occur and define how the system should behave in those cases.

Correctness:

- Actors: The actors are correctly identified.

# ECHO GROUP

- Preconditions: The preconditions are incomplete. It is important to report if the patient has already installed the Mobile App and if the Sensor Device is available.
- Postconditions: The postconditions are accurate and correctly describe the state of the system after the use case has been executed.
- Basic Flow: The basic flow is not atomic. The step "The Sensor Device sends the raw data to the Mobile" should be split into two steps. The first step is "The Sensor Device sends the raw data to the Mobile App," and the second step is "The Mobile App receives the raw data from the Sensor Device." It is also missing the step to validate the measurement data before saving. The recommended change is to add a step in the basic flow to validate the measurement data before saving. For example: "The Mobile App validates the measurement data received from the Sensor Device before sending the request to save the data to the System."
- Alternative Flows: There are no alternative flows.
- Error Handling: The error handling is missing. It is essential to consider potential errors that may occur, such as the Sensor Device malfunctioning or the Mobile App losing connectivity, and define how the system should behave in those cases.

## Clarity:

- The use case is easily understandable for all stakeholders.
- There is no extraneous information or jargon that could impede understanding.
- The terminology used in the use case is consistent, and there is no discrepancy in the steps' flow.

However, it would be helpful to clarify the interactions between the Mobile App, the Sensor Device and the system.

## 4.3 Second ChatGPT suggestions

### Completeness:

- Actors: The actors are correctly identified and reported.
- Preconditions: The preconditions are reported but they are incomplete. It is necessary to add a precondition that the Patient has the kit with the instrument to measure the vital sign.
- Postconditions: The postconditions are reported and complete.
- Basic Flow: The basic flow is complete, but there is no mention of input data validation. It is necessary to add a validation step to check the raw data received from the Sensor Device.

# ECHO GROUP

- Alternative Flows: There are no alternative flows reported, and this is correct.
- Error Handling: The error handling is complete and covers all potential errors.

## Correctness:

- Actors: The actors are correctly identified.
- Preconditions: The preconditions are not accurate and do not identify the success criteria for the main scenario. It is necessary to add a precondition that the Patient has the kit with the instrument to measure the vital sign.
- Postconditions: The postconditions accurately describe the state of the system after the use case has been executed.
- Basic Flow: The basic flow is correctly ordered, but it is not atomic. The steps 2 and 3 should be split into smaller steps to improve atomicity.
  - Step 2: The Sensor Device sends the raw data to the Mobile App.
  - Step 3: The Mobile App receives the raw data from the Sensor Device, validates the measurement data, and sends the request to save the data to the System.
- Alternative Flows: There are no alternative flows reported, and this is correct.
- Error Handling: The error handling is accurate and covers all potential errors that may occur.

## Clarity:

- The use case is easily understandable for all stakeholders, including developers and non-developers. There is no extraneous information or jargon that could impede understanding.
- The use case is consistent in terms of logical sequence and terminology consistency.

## 5 Use case 5

ID: CPM:US2.001

Title: List all alert parameters

Actors: Doctor

Preconditions:

- The user selects the "Thresholds" in a patient's medical record

## Main scenario:

1. The user selects the "List all vitals" button.
2. The system searches and displays the list of vitals.
3. The user selects the desired vital sign.

# ECHO GROUP

4. The system searches and displays the list of alerts related to the given vital parameter. The list is sortable by "Name".

## 5.1 ChatGPT last improved version

ID: CPM:US2.001

Title: List all alert parameters

Actors: Doctor

### Preconditions:

- The Doctor is logged in to the system and has accessed a patient's medical record.
- The Doctor has navigated to the "Thresholds" section of the patient's medical record.

Main scenario:

1. The Doctor selects the patient's medical record.
2. The Doctor navigates to the "Thresholds" section of the patient's medical record.
3. The Doctor selects the "List all vital signs" button.
4. The system validates the input.
5. The system searches for and retrieves the list of vital signs associated with the patient, including each vital sign's value, units, and timestamp.
6. The system displays the list of vital signs to the Doctor.
7. The Doctor selects the desired vital sign from the list of vital signs.
8. The system validates the input.
9. The system searches for and retrieves the list of alerts related to the selected vital sign, sorted by "Name".
10. The system displays the list of alerts to the Doctor.
11. The Doctor reviews the list of alerts and takes any necessary action.

### Alternative Flow:

- 3a. The Doctor decides to exit and presses the "Exit" button.
- 4a. The system returns to the Homepage.

### Error Handling:

- If the Doctor enters an invalid vital sign, the system displays an error message and prompts the Doctor to enter a valid vital sign.
- If the system encounters an error while searching for vital signs or alerts, an error message is displayed to the Doctor.

**Postconditions:** The Doctor has reviewed the list of alerts related to the selected vital sign and has taken any necessary action.

## 5.2 First ChatGPT suggestions

Completeness:

- Actors: The actor "Doctor" is reported.

# ECHO GROUP

- Preconditions: The precondition is missing. It should be specified that the user is logged in to the system and has access to the patient's medical record.
- Postconditions: The postcondition is missing. It should be specified what happens after the user selects the desired vital sign.
- Basic Flow: The basic flow is incomplete. It should include the steps for the user to select the patient's medical record and navigate to the "Thresholds" section. Input data validation is not reported in the basic flow.
- Alternative Flows: No alternative flows are reported.
- Error Handling: Error handling is missing. It should be specified what happens if the system encounters an error while searching for vitals or alerts.

## Correctness:

- Actors: The actor is correctly identified as "Doctor."
- Preconditions: The preconditions do not accurately identify the success criteria for the main scenario of the use case. It should be specified that the user has selected a patient's medical record and navigated to the "Thresholds" section.
- Postconditions: The postcondition is missing. It should be specified what happens after the user selects the desired vital sign.
- Basic Flow: The basic flow is not atomic. Step 1 should be split into two separate steps: a) select the patient's medical record, and b) navigate to the "Thresholds" section. Step 2 should include input data validation to ensure that the list of vitals is not empty. Step 4 should be split into two separate steps: a) search for alerts related to the given vital parameter, and b) display the list of alerts.
- Alternative Flows: No alternative flows are reported. It could be helpful to add an alternative flow to specify what happens if the user decides to cancel the operation.
- Error Handling: Error handling is missing. It should be specified what happens if the system encounters an error while searching for vitals or alerts.

## Clarity:

- The use case is relatively clear and understandable for all stakeholders. However, it could benefit from additional details, such as specifying the type of medical record being accessed and clarifying the purpose of the "Thresholds" section.
- The use case is inconsistent in its use of terminology. It would be helpful to change "sortable" to "sorted".



# ECHO GROUP

## 5.3 Second ChatGPT suggestions

### Completeness:

- Actors: The actor is reported.
- Preconditions: The preconditions are reported.
- Postconditions: The postconditions are reported.
- Basic Flow: The basic flow is complete. However, it could be improved by including more details about what happens when the Doctor selects a vital sign from the list. Does the system display the vital sign's value, units, and timestamp? Is the list of alerts displayed in a new window or on the same page? Additionally, it is unclear whether input data validation is reported in the basic flow.
- Alternative Flows: None reported.
- Error Handling: The error handling is incomplete as it does not specify what happens if the Doctor enters an incorrect input, such as an invalid vital parameter.

### Correctness:

- Actors: The actor is correctly identified.
- Preconditions: The preconditions accurately identify the success criteria for the main scenario of the use case.
- Postconditions: The postconditions accurately describe the state of the system after the use case has been executed.
- Basic Flow: The basic flow is correctly ordered, but it is not atomic. To improve atomicity, the steps could be split as follows:
  1. The Doctor selects the patient's medical record.
  2. The Doctor navigates to the "Thresholds" section of the patient's medical record.
  3. The Doctor selects the "List all vitals" button.
  4. The system validates the input and displays the list of vitals associated with the patient.
  5. The Doctor selects the desired vital sign from the list of vitals.
  6. The system validates the input and displays the list of alerts related to the selected vital parameter.
  7. The Doctor reviews the list of alerts and takes any necessary action.
- Alternative Flows: None reported.
- Error Handling: The error handling is incomplete and should be updated to cover all potential errors that may occur.

### Clarity:

- The use case is easily understandable for all stakeholders, including developers and non-developers. However, it could be improved by including more details about what happens when the Doctor selects a vital sign from the list. Additionally, "vitals" could be replaced with "vital signs" for clarity.