Requirements Engineering Document

Prioritized Requirements

Essential:

1. A log in system must exist to give access to doctors and nurses.
2. A patient profile creation system or mechanism that either a doctor or nurse can use.
3. The user should have the ability to change the default vital sign safe limits in the system.
4. The system should be able to read data from the sensors (input data stream) of a patient
5. The system should display an emergency and make a sound whenever the vital signs of a patient reach the limit.
6. The user should have the ability to stop the system from monitoring the patient.
7. The system should convert all of the data being read from the sensors to appropriate units.
8. The system should log all of the data being read from the system into a patient's separate profile.
9. The system should display all of the data being read from the sensors onto the screen.
10. A doctor or nurse should have the ability to see the patient's profile on the screen.

Desirable:

1. The doctor should be the only person with the ability to change the vital sign safe limits
2. The system should intelligently create enough space for each patient profile to log data since not every patient's hospital stay will be of the same length.

Optional:

1. The system should automatically log off a user after a certain amount of time.
2. The system should display a warning and make a sound when the vital signs of a patient come within 20% of the vital sign limits.

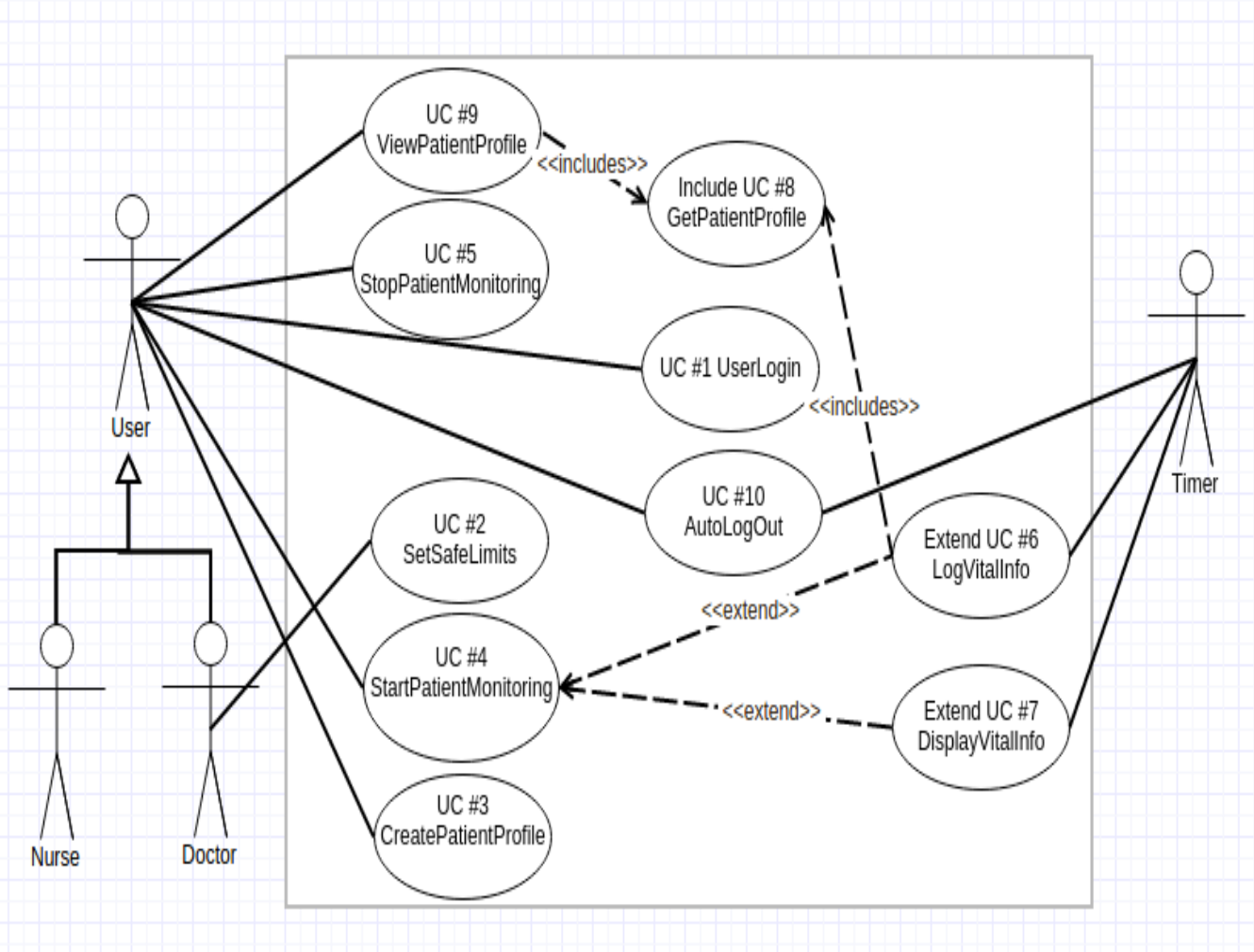
Functional Requirements

1. A patient profile creation system or mechanism that creates a profile for a patient with an associated name and patient ID number.
2. The system should give the user the ability to change default vital sign safe limits with new inputs.
3. The system should be able to read data from the sensors (input data stream) of a patient.
4. The system should display an emergency and make a sound whenever the vital signs of a patient reaches or exceeds the limit stored on the system.
5. The user should have the ability to stop the system from further monitoring the patient.
6. The system should convert all of the data being read from the sensors to appropriate units (mm Hg for blood pressure, bpm for heart rate, and degrees Celsius for temperature).
7. The system should log all of the data being read from the system into a patient's separate profile after every three seconds.
8. The system should display all of the data being read from the sensors onto the screen after every three seconds.
9. A doctor or nurse should have the ability to see the patient's profile on the screen.
10. The system should display a warning and make a sound when the vital signs of a patient come within 20% of the vital sign limit.

Non-Functional Requirements

1. The system should know when a doctor logs on to the system and only allow them to change the vital sign safe limits. (security)
2. When a user is logging in and the username and/or password is wrong, the system should not inform the user which of the two is wrong in order to prevent guessing. (security)
3. The system should automatically log a user out after five minutes of user inactivity. (security)
4. The system should intelligently create enough space for the patient's profile when vital sign data is being inputted into it. (robustness)
5. The system should not allow the doctor to exceed a certain pre-set maximum range for each of the vital signs when he/she is inputting new limits. (reliability)
6. The system should prevent multiple profiles for a patient by checking to see if the patient ID exists on the system when the user is creating a new profile (reliability).
7. The system should signal a warning and sound when there is no data coming from the sensors (reliability).

Use Case Diagram



Use Case #1 – UserLogin

Brief Description:
The user enters his/her credentials to login to the system.

Primary Actors:
User (Doctors and Nurses).

Secondary Actors:
None.

Triggers:
User selects operation in system to login.

Main Flow:
1. Use case starts when user selects “login” in system.
2. The system requests user to input his/her username and password.
3. The system validates user's username and password.
4. The system logs user in.

Alternative Flows:
1.1 UserCancel
1.2 InvalidLogin

Preconditions:
None

Success Postconditions (success guarantee):
User is logged in to system.

Failure Postconditions (minimal guarantee):
User is not logged in to system.

Extension Points:
None.

Alternative Flow #1.1 – UserLogin: UserCancel

Brief Description:
User cancels login process.

Primary Actors:
User (Doctor/Nurse).

Secondary Actors:
None.

Triggers:
User selects operation in system to cancel during login process.

Alternative Flow:
1. The alternative flow begins anytime customer selects “cancel” during UC#1 (UserLogin).
2. The user is not logged in and the system returns to the first menu.

Preconditions:
None.

Postcondition:
User is not logged in to system.

Extension Points:
None.

Alternative Flow #1.2 – UserLogin: InvalidLogin

Brief Description:
The system notifies user that he/she has entered an invalid username or password.

Primary Actors:
User (Doctor/Nurse)

Secondary Actors:
None.

Triggers:
User enters a username/password that system cannot validate.

Alternative Flow:
1. The alternative flow starts after step 2 of the main flow in UC#1 (UserLogin).
2. System notifies user that he/she has entered an invalid user name and/or password.
3. Alternative flow returns to step 2 in UC#1(UserLogin).

Preconditions:
User has entered an invalid username or password.

Postconditions:
System displays notification of wrong username and/or password

Extension Points:
None.

Use Case #2 – SetSafeLimits

Brief Description:
Doctor inputs the safe limits for the patient's vital signs (heart rate, temperature and blood pressure) into the system.

Primary Actors:
Doctor

Secondary Actors:
None.

Triggers:
Doctor selects operation in menu to input safe limits for patient vital signs.

Main Flow:
1. Use Case starts when Doctor selects “Change vital limits” in menu.
2. The system requests user to input vital sign limits for blood pressure, heart rate, and temperature.
3. Doctor enters new vital sign limits into system.
4. System checks that that none of the doctor's inputs is out of allowable limit range.
5. Previously stored vital signs limits are overwritten with limits doctor enters.

Alternative Flows:
2.1 CancelLimitInput
2.2 InvalidLimitInput

Preconditions:
Doctor must be logged into system.

Success Postconditions (success guarantee):
Previously stored vital sign safe limits are overwritten with limits the doctor enters.

Failure Postconditions (minimal guarantee):
Previously stored vital sign safe limits are retained.

Extension Points:
None.

Alternative Flow #2.1 – CancelLimitInput

Brief Description:
The doctor cancels vital signs limit input and previously stored safe limits are retained by the system.

Primary Actors:
Doctor

Secondary Actors:
None.

Triggers:
Doctor cancels vital signs limit input anytime during UC#2(SetSafeLimits).

Alternative Flow:
1. Alternative flow starts whenever doctor selects “cancel” anytime during UC#2 (SetSafeLimits).
2. The system retains previously stored vital signs limits.

Preconditions:
Doctor must be logged into system.

Postconditions:
Previously stored safe limits for vital signs are retained.

Extension Points:
None.

Alternative Flow #2.2 – InvalidLimitInput

Brief Description:
The system gives a warning when the user gives an input that falls out of the range for at least one of the vital signs (blood pressure, heart rate, temperature).

Primary Actors:
Doctor

Secondary Actors:
None.

Triggers:
Doctor inputs invalid vital sign limits after step 3 of UC#2 (SetSafeLimits).

Alternative Flow:
1. Alternative flow starts when Doctor enters at least one vital sign limit that falls out of allowed range after step 3 of UC#2 (SetSafeLimits).
2. System displays warning that inputs fall out of allowed range.
3. Alternative flow returns to step 2 of UC#2 (SetSafeLimits).

Preconditions:
Doctor must be logged into system.

Postconditions:
System displays warning that inputs exceed allowable range.

Extension Points:
None.

Use Case #3 – CreatePatientProfile

Brief Description:
The system creates a new profile for patient.

Primary Actors:
User (Doctor or Nurse).

Secondary Actors:
None.

Triggers:
User selects operation in system to create new profile for patient.

Main Flow:
1. Use case starts when user selects “Create Patient Profile”.
2. User inputs patient name and ID number.
3. The system creates a profile for patient with file to log patient medical information.

Alternative Flows:
3.1. Pre-ExistingProfile
3.2 CancelProfileCreation

Preconditions:
User must be logged in to system.

Success Postconditions (success guarantee):
Patient has a profile associated with his/her patient ID.

Failure Postconditions (minimal guarantee):
None.

Extension Points:
None.

Alternative Flow #3.1 – Pre-ExistingProfile

Brief Description:
System informs user that patient already has a profile in system.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
None.

Triggers:
User enters a patient ID that is already associated with a profile in system during UC#3 (CreatePatientProfile).

Alternative Flows:
1. Alternative flow starts when user enters a patient ID that is already associated with a profile in the system during UC#3 (CreatePatientProfile).
2. System informs user that patient already has a profile.

Preconditions:
Profile associated with patient ID must exist on system.

Postconditions:
Patient has a profile on system.

Extension Points:
None.

Alternative Flow #3.2 – CancelProfileCreation

Brief Description:
The user cancels patient profile creation.

Primary Actors:
User (Doctor or nurse).

Secondary Actors:
None.

Triggers:
Doctor cancels patient profile creation during UC#3 (CreatePatientProfile).

Alternative Flow:
1. Alternative flow starts whenever doctor selects “cancel” during UC#3 (CreatePatientProfile).
2. The system cancels creating a profile for the patient and returns to menu (post-login).

Preconditions:
User must be logged into system.

Postconditions:
None.

Extension Points:
None.

Use Case #4 – StartPatientMonitoring

Brief Description:
This use case begins the monitoring of the patient's vital signs and converts incoming data into appropriate units.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
None.

Triggers:
User selects operation in system to begin patient monitoring.

Basic Flow:
1. The use case starts when the user selects “Monitor Patient”.
2. System asks user for patient's ID number.
3. User inputs patient ID number.
4. System validates patient ID number.
5. System starts reading patient's vital signs from sensors(input data stream).
6. The system converts all incoming patient vital input data into appropriate units (mm Hg for blood pressure, bpm for heart rate, and degrees celsius for temperature).
Extension point: UC#6 (LogVitalInfo)
Extension point: UC#7 (DisplayVitalInfo)

Alternative Flows:
4.1 NoVitalInput
4.2 WrongIDNumber
4.3 VitalSignEmergency
4.4 VitalSignWarning

Preconditions:
User is logged in to system.

Success Postconditions (success guarantee):
System is reading patient's vital signs from the sensors (input data stream).

Failure Postconditions (minimal guarantee):
System is not reading patient's vital signs from sensors and has given an appropriate notification.

Extension Points:
UC#6 LogVitalInfo
UC #7 DisplayVitalInfo

Alternative Flow #4.1 - NoVitalInput

Brief Description:
System is not receiving any data or is disconnected from sensors (vital input data) and warns user.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
None.

Triggers:
No data being read from sensors after step 5 of UC#4 (StartPatientMonitoring).

Alternative Flows:
1. The alternative flow when the system doesn't detect any data or is disconnected from sensors after step 5 of StartPatientMonitoring use case.
2. System displays an error to user that sensors are disconnected.

Preconditions:
None.

Postconditions:
System displays warning that sensors are disconnected or outputting no data.

Extensions:
None.

Alternative Flow #4.2 - WrongIDNumber

Brief Description:
System warns user that patient ID number he/she has given does not exist.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
None.

Triggers:
User inputs a patient ID number not found in system after step 4 of UC#4 (StartPatientMonitoring).

Alternative Flow:
1. The alternative flow starts when the user enters an invalid patientID number after step 4 of StartPatientMonitoring use case.
2. System displays warning to user that the ID number doesn't exist and returns to step 2 of StartPatientMonitoring use case.

Preconditions:
Patient ID number not found on system.

Postconditions:
System is display error that patientID number is wrong.

Extensions:
None.

Alternative Flow #4.3 - VitalSignEmergency

Brief Description:
System display emergency and alarm sound user that the vital signs limits have fallen out of vital sign limit range.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
None.

Triggers:
Data being read from sensors has fallen out of the system's vital sign limit range.

Alternative Flow:
1. The alternative flow starts whenever the input data from the sensors falls out of the system's vital sign limit range during UC#4 (StartPatientMonitoring).
2. System displays emergency warning and alarm sound that patient's vital signs have fallen out of the allowable range.

Preconditions:
Patient vital signs are being monitored.

Postconditions:
System is displaying an emergency warning and alarm sound that patient's vital signs are out of allowable range.

Extensions:
None.

Alternative Flow #4.4 - VitalSignWarning

Brief Description:
System displays warning display and sound that patient's vital signs are within 20% of either the upper or lower limits of system's vital sign range.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
None.

Triggers:
Data being read from sensors has fallen out of the system's vital sign limit range.

Alternative Flow:
1. The alternative flow starts whenever the input data from the sensors reaches 20% of the upper or lower limit of the system's vital sign range during UC#4 (StartPatientMonitoring).
2. System displays warning and sound.

Preconditions:
Patient's vital signs are being monitored.

Postconditions:
System is displaying an emergency warning and sound that patient's vital signs re within 20% of either upper or lower limits of system's vital sign range.

Extensions:
None.

Use Case #5 – StopPatientMonitoring

Brief Description:
This use case stops the monitoring of the patient's vital signs.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
None.

Triggers:
User selects operation in system to stop patient monitoring.

Basic Flow:
1. The use case starts when the user selects “Stop Patient Monitoring”.
2. System stops reading patient's vital signs from sensors (input data stream).

Alternative Flows:
None.

Preconditions:
User is logged into system.
System is monitoring patient.

Success Postconditions (success guarantee):
System is not reading patient's vital signs.

Failure Postconditions (minimal guarantee):
None.

Extension Points:
None.

Extend Use Case #6 – LogVitalInfo

Brief Description:
The system logs patient's vital information into the patient's profile while it is monitoring patient.

Primary Actors:
User (Doctor and Nurse).

Secondary Actors:
Timer.

Triggers:
User selects operation to begin patient monitoring.

Main Flow:
1. The use case begins after step 2 in UC#4 (StartPatientMonitoring).
2. Include UC#8 (GetPatientPortfolio).
3. For every three seconds in Timer:
   3.1 System inputs patient's real-time vital information from sensor (input data stream) into patient's profile.

Alternative Flows:
6.1. PatientProfileFull

Preconditions:
1. System is monitoring patient's vital signs.

Success Postconditions (success guarantee):
Patient's vital signs are being recorded into patient profile

Failure Postconditions (minimal guarantee):
Patient's vital signs are being recorded into patient profile.

Extensions:
None

Alternative Flow #6.1 - PatientProfileFull

Brief Description:
Patient's profile has reached full capacity and system creates more space for profile to log information.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
Timer

Triggers:
Patient's profile is full during the LogVitalInfo(UC #6) use case.

Alternative Flow:
1. The alternative flow starts whenever the patient's profile has reached it's maximum capacity during UC#6 (LogVitalInfo).
2. System allocates additional space for patient's profile.
3. Alternative flow returns to step 4 of UC#6 (LogVitalInfo).

Preconditions:
Patient's vital signs are being monitored.

Postconditions:
Patient's vital signs are being recorded into patient profile.

Extensions:
None.

Extend Use Case #7 - DisplayVitalInfo

Brief Description:
The system displays patient's vital information while patient is being monitored.

Primary Actors:
User (Doctor and Nurse).

Secondary Actors:
Timer.

Triggers:
User selects operation to begin patient monitoring.

Main Flow:
1. The use case begins after step 5 in UC#4 (StartPatientMonitoring).
2. For every three seconds in Timer:
   2.1 System displays patient's real-time vital information from sensor (input data stream).

Alternative Flows:
7.1. NoDisplayConnected

Preconditions:
System is monitoring patient's vital signs.

Success Postconditions (success guarantee):
Patient's vital signs are displayed on screen

Failure Postconditions (minimal guarantee):
System is monitoring patient and giving sound warning that display is disconnected.

Extensions:
None.

Alternative Flow #7.1 - NoDisplayConnected

Brief Description:
System sounds warning that display is connected while patient is being monitored.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
Timer

Triggers:
Display is disconnected during UC#7 (DisplayVitalInfo).

Alternative Flow:
1. The alternative flow starts whenever the display is disconnected from the system during UC#7 (DisplayVitalInfo).
2. System sounds warning that display is disconnected and continues to monitor patient.

Preconditions:
Patient's vital signs are being monitored.

Postconditions:
System is monitoring patient and giving sound warning that display is disconnected.

Extensions:
None.

Include Use Case #8 – GetPatientProfile

Brief Description:
System retrieves patient's profile

Primary Actors:
User (Doctor or Nurse).

Secondary Actors:
None.

Triggers:
User is viewing patient profile or system is logging info into patient profile.

Main Flow:
1. The include use case starts when either the user is viewing a patient's profile or the system is logging the patient's vital information into to the profile.
2. System uses patient's ID number to retrieve patient's profile.

Alternative Flows:
8.1 PatientProfileNotFound.

Preconditions:
None.

Success Postconditions (success guarantee):
Patient's profile has been retrieved.

Failure Postconditions (minimal guarantee):
System displays error that patient's profile is not on the system.

Extensions:

Alternative Flow #8.1 - PatientProfileNotFound

Brief Description:
System displays error that patient's profile is not on the system.

Primary Actors:
User (Doctor or Nurse)

Secondary Actors:
None.

Triggers:
System cannot find profile associated with patient's ID.

Alternative Flow:
1. The alternative flow starts after step 2 of UC#8 (GetPatientProfile) when the system cannot find a profile associated with the patient's ID.
2. System displays error that patient's profile is not on system.

Preconditions:
None.

Postconditions:
System is displaying error that patient's profile is not on the system.

Extensions:
None.

Use Case #9 - ViewPatientProfile

Brief Description:
The system displays patient's profile.

Primary Actors:
User (Doctor or Nurse).

Secondary Actors:
None.

Triggers:
User selects operation on system to display patient's profile.

Main Flow:
1. The use case begins when the user selects “View Patient Profile” in menu.
2. Include UC #8(GetPatientProfile).
3. System displays patient's profile.

Alternative Flows:
None.

Preconditions:
User must be logged in to system.

Success Postconditions (success guarantee):
User is able to view patient's profile.

Failure Postconditions (minimal guarantee):
None.

Extensions:
None.

Use Case #10 – AutoLogOut

Brief Description:
System automatically logs user out after 5 minutes of inactivity.

Primary Actors:
Timer

Secondary Actors:
User (Doctor or Nurse).

Triggers:
User has been inactive for more than 5 minutes.

Main Flow:
1. The use case starts when 5 minutes of inactivity has elapsed in system.
2. System automatically logs user off.

Alternative Flows:
None.

Preconditions:
User must be logged in and inactive for 5 minutes.

Success Postconditions (success guarantee):
User is logged off by system.

Failure Postconditions (minimal guarantee):
None.

Extensions:
None.