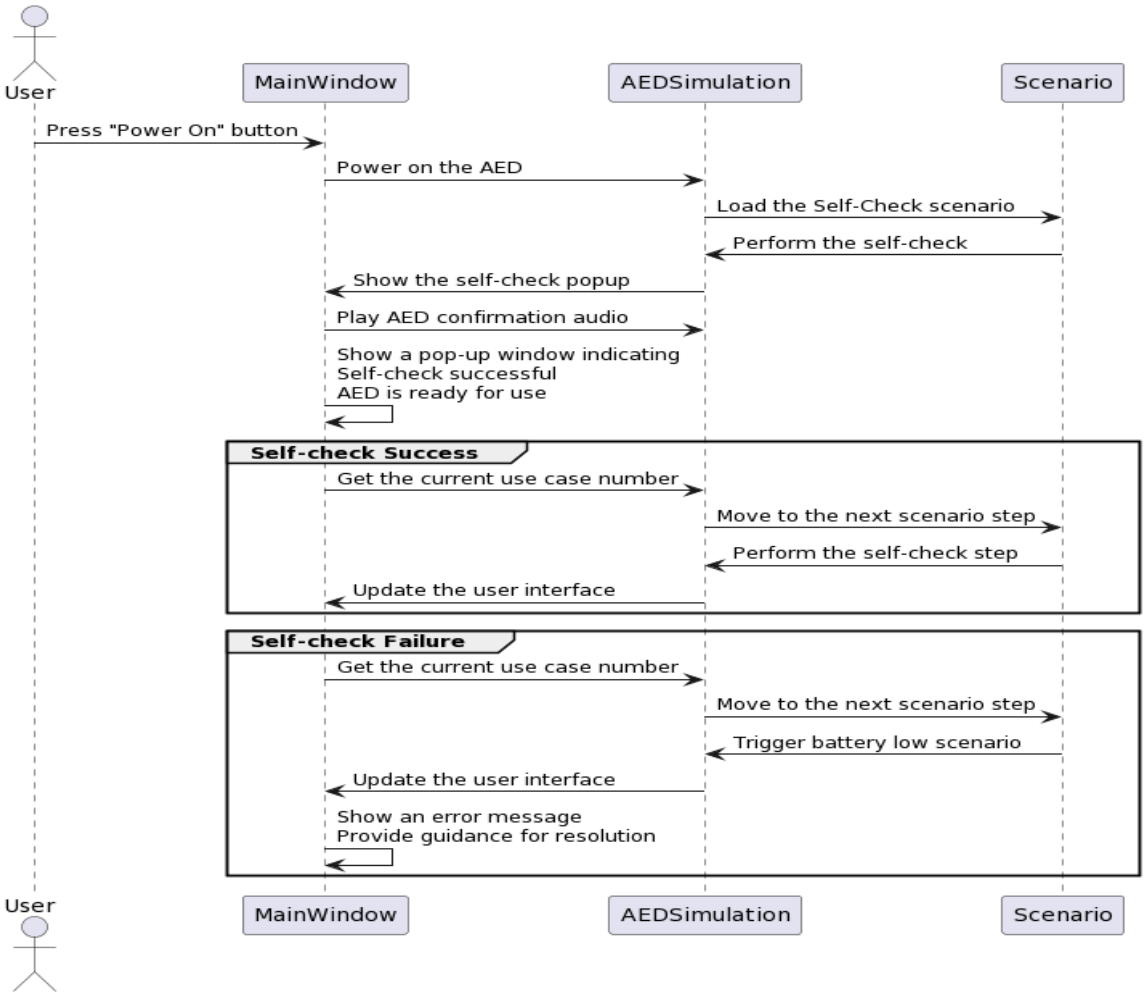
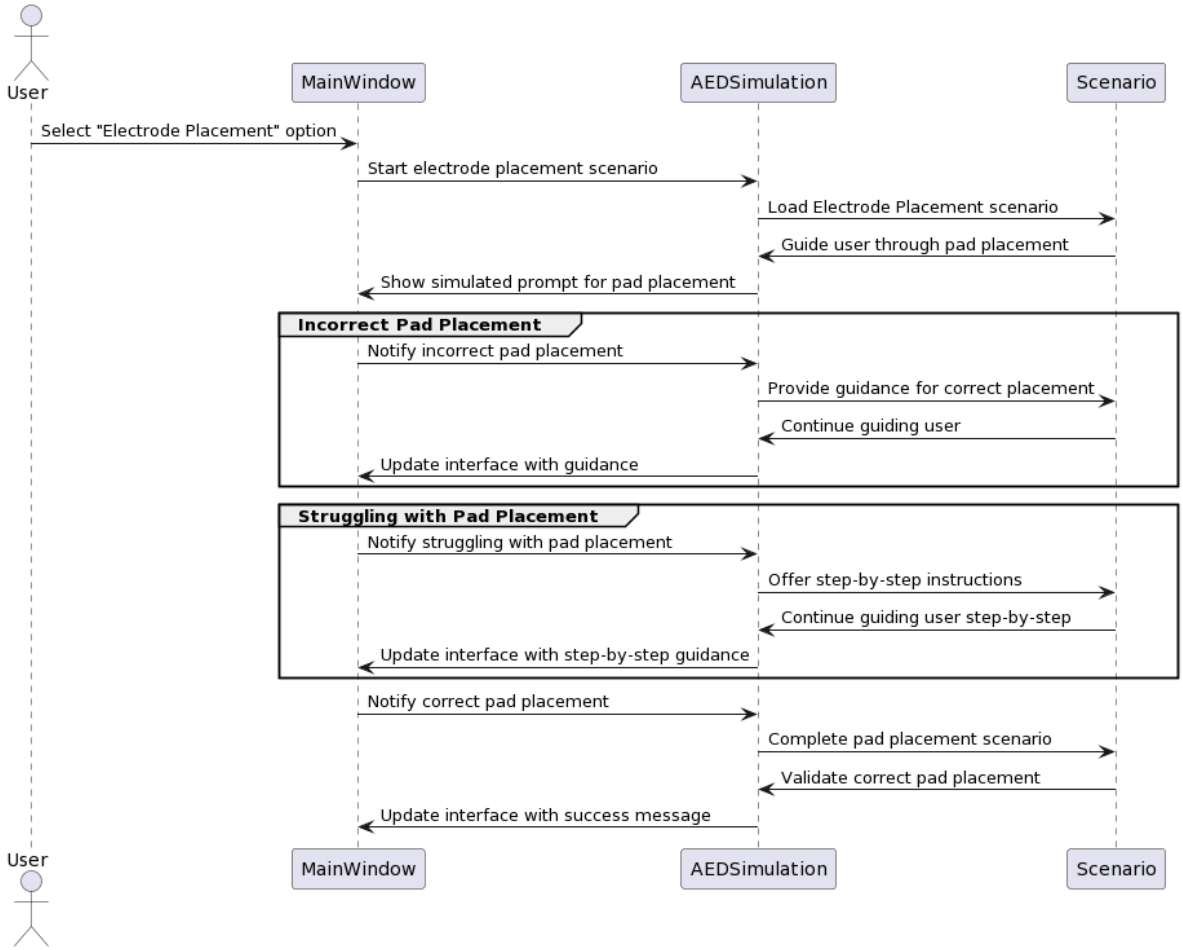
Use case 1: Power On

1. **Name:** Power On
2. **Actor:** User
3. **Pre-conditions:** AED is turned off
4. **Main Success Scenario:**
    a. User presses the “Power On” button
    b. A pop-up window appears, informing the user that the machine has done a self-check, and everything is operational.
    c. AED audio plays, confirming ready for use.
5. **Extensions:**
    a. If the AED encounters issues during the self-check(step 4b), an error message is provided and guidance for resolution.
6. **Success Guaranteed:** AED is powered on and the user receives confirmation of successful complete self-check and operational status.



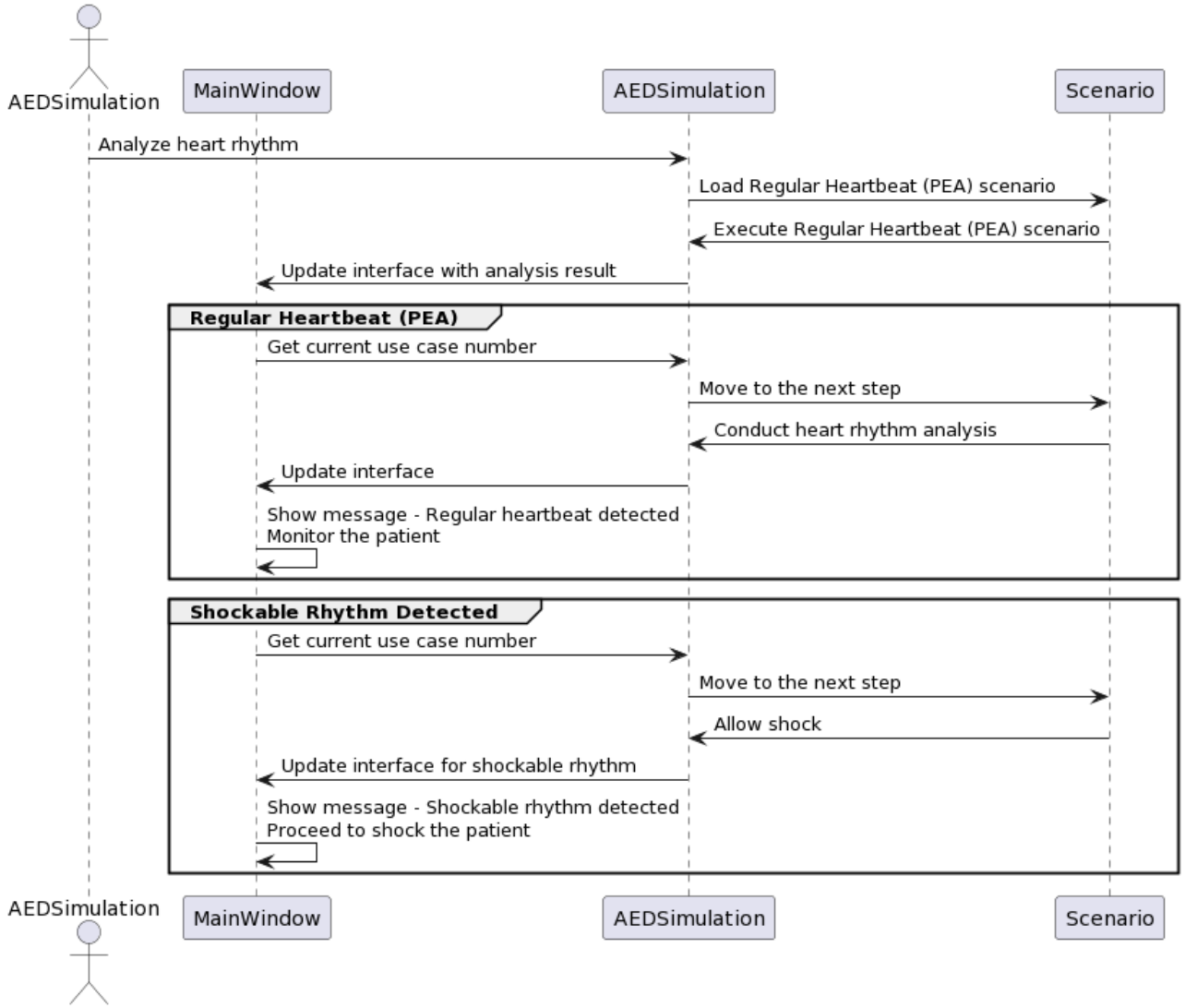
Use case 2: Pad Already Placed on Patient

1. **Name:** Pad Placement
2. **Actor:** User
3. **Pre-conditions:** AED is turned on and passes self-check
4. **Main Success Scenario:**
    a. User selects the “Electrode Placement” option
    b. AED provides a simulated prompt, guiding the user through the process of placing the pads accurately on the patient's bare chest.
5. **Extensions:**
    a. If the user places the pad incorrectly(step 4b), the AED provides guidance for correct placement.
    b. If the user is still struggling with pad placements(step 4b), the AED offers a step-by-step instruction until the pads are placed properly.
    c. If there is a loss of connection during the process, it goes back to placing pads.
6. **Success Guaranteed:** Electrode pads are placed correctly on the patients chest, to ensure accurate readings.



Use case 3: Regular Heartbeat Reading(PEA)

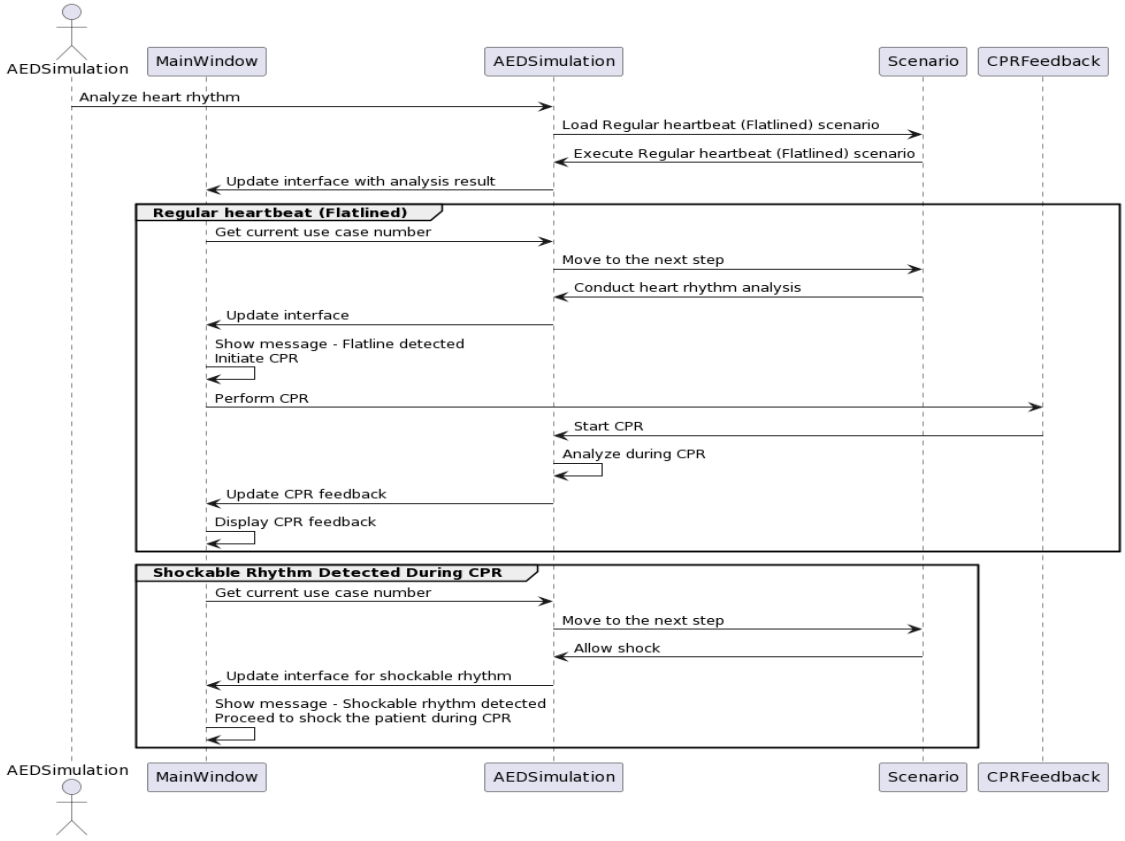
1. **Name:** Regular Heartbeat Reading(PEA)
2. **Actor:** AED simulation
3. **Pre-conditions:** AED is turned on and electrode pads are placed properly
4. **Main Success Scenario:**
    a. AED analyzes the patient's heart rhythm and detects a regular heartbeat(PEA).
    b. The AED provides audio instructions to monitor the patient while waiting for help.
5. **Extensions:**
    a. If the AED detects a shockable rhythm(step 4a), it moves to the appropriate scenarios(e.g. irregular heartbeat VF or VT).
    b. If there is a loss of connection during the process, it goes back to placing pads.
6. **Success Guaranteed:** AED recognizes the regular heartbeat and instructs the user to monitor the patient.



Use case 4: Regular heartbeat (Flatlined)

1. **Name:** Regular Heartbeat (Flatlined)
2. **Actor:** AED Simulation
3. **Pre-conditions:** AED is turned on and electrode pads are placed properly.
4. **Main Success Scenario:**
    a. AED analyzes the patient's heart rhythm and detects a flatline.
    b. The AED provides audio and visual prompts indicating the need for CPR.
    c. The AED displays helpful tips, guiding the user on the proper chest compression depth(2 to 4 inches) and providing real-time feedback.
    d. If compression depth is insufficient, the AED displays “Bad compression” and guides the user on improvement technique(s).
    e. After CPR initiation, the AED continues to analyze for shocks, providing real-time feedback and instructions.

5. **Extensions:**
    a. If the AED detects a shockable rhythm during analysis(step 4a), it transitions to the appropriate scenario(e.g. irregular heartbeat VF or VT).
    b. If there is a loss of connection during the process, it goes back to placing pads.
6. **Success Guaranteed:** AED recognizes flatlined rhythm, guides the user through CPR and provides feedback.

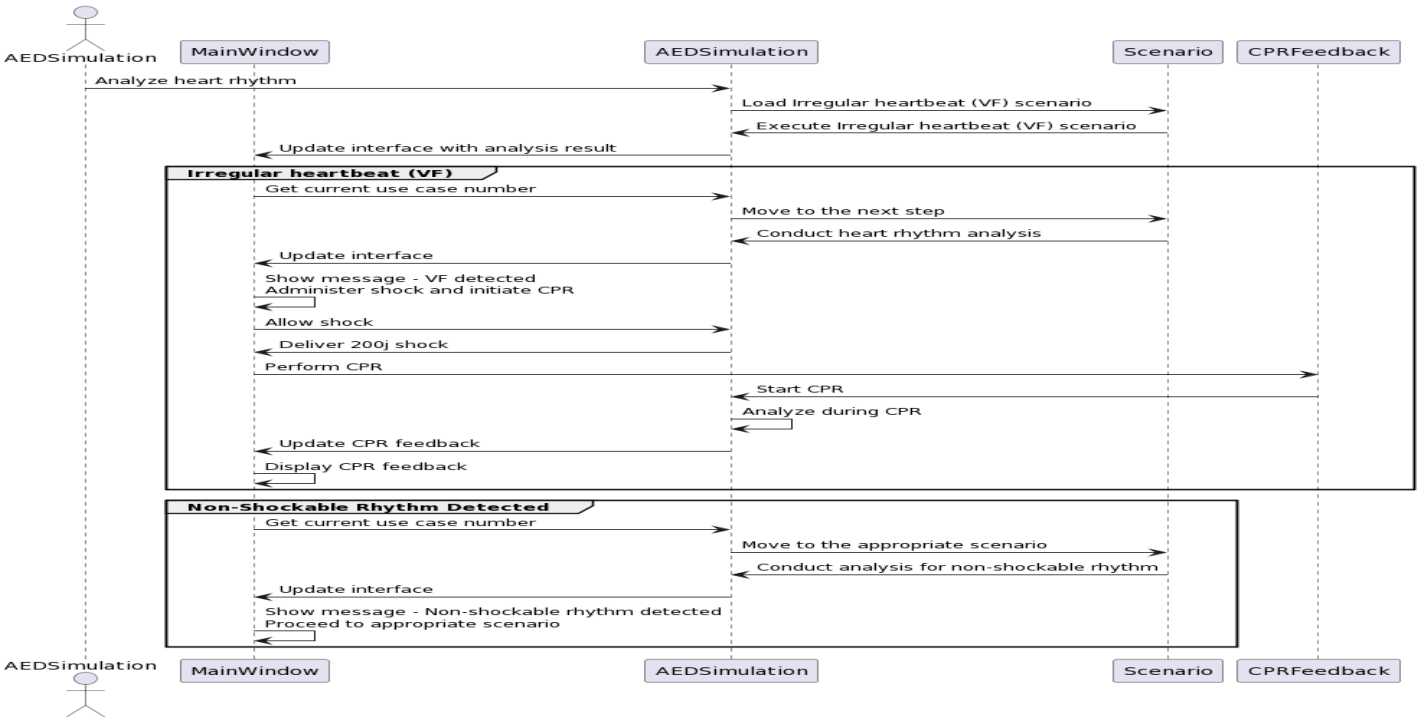


Use case 5: Irregular heartbeat (Ventricular Fibrillation(VF))

1. **Name:** Irregular Heartbeat (VF)
2. **Actor:** AED Simulation
3. **Pre-conditions:** AED is turned on and electrode pads are placed properly.
4. **Main Success Scenario:**
    a. AED analyzes the patient's heart rhythm and detects VF.
    b. The AED provides audio and visual prompts indicating the availability of shocks.
    c. User initiates shock delivery, AED administers 200j shock(if administering more than 1 shock, then the AED should administer 300j shock from there on).

d. The AED prompts the user to perform CPR.
e. The AED continues to analyze for additional shocks, delivering randomized 1-3 shocks if needed.
f. If the patient stabilizes, the AED transitions to CPR automatically.

5. **Extensions:**
    a. If the AED detects a non-shockable rhythm during analysis(step 4a), it transitions to the appropriate scenario(e.g. regular heartbeat flatlined).
    b. If there is a loss of connection during the process, it goes back to placing pads.
6. **Success Guaranteed:** AED recognizes VF, delivers shocks, guides the user through transition to CPR if stabilization occurs.

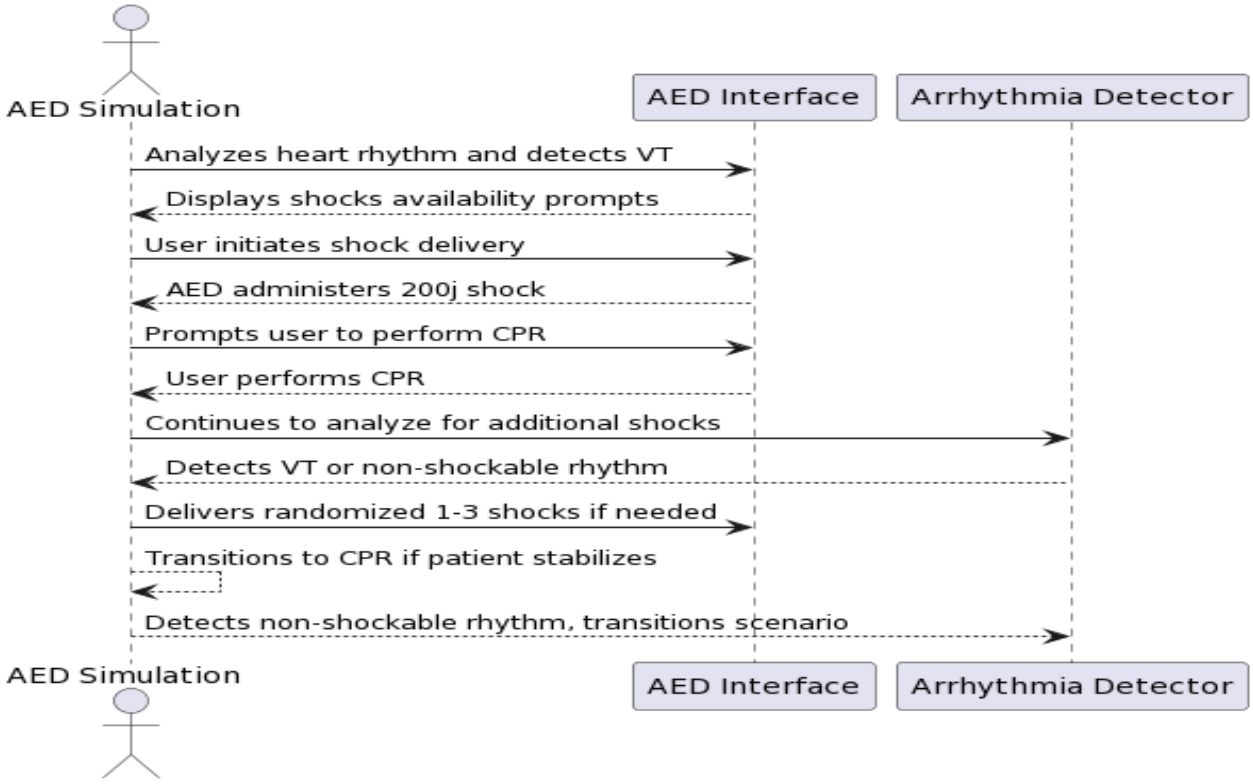


Use case 6: Irregular heartbeat (Ventricular Tachycardia(VT))

1. **Name:** Irregular Heartbeat (VT)
2. **Actor:** AED Simulation
3. **Pre-conditions:** AED is turned on and electrode pads are placed properly.
4. **Main Success Scenario:**
    a. AED analyzes the patient's heart rhythm and detects VT.
    b. The AED provides audio and visual prompts indicating the availability of shocks.
    c. User initiates shock delivery, AED administers 200j shock(if administering more than 1 shock, then the AED should administer 300j shock from there on).

d. The AED prompts the user to perform CPR.
e. The AED continues to analyze for additional shocks, delivering randomized 1-3 shocks if needed.
f. If the patient stabilizes, the AED transitions to CPR automatically.

5. **Extensions:**
    a. If the AED detects a non-shockable rhythm during analysis(step 4a), it transitions to the appropriate scenario(e.g. regular heartbeat flatlined).
    b. If there is a loss of connection during the process, it goes back to placing pads.
6. **Success Guaranteed:** AED recognizes VT, delivers shocks, guides the user through transition to CPR if stabilization occurs.

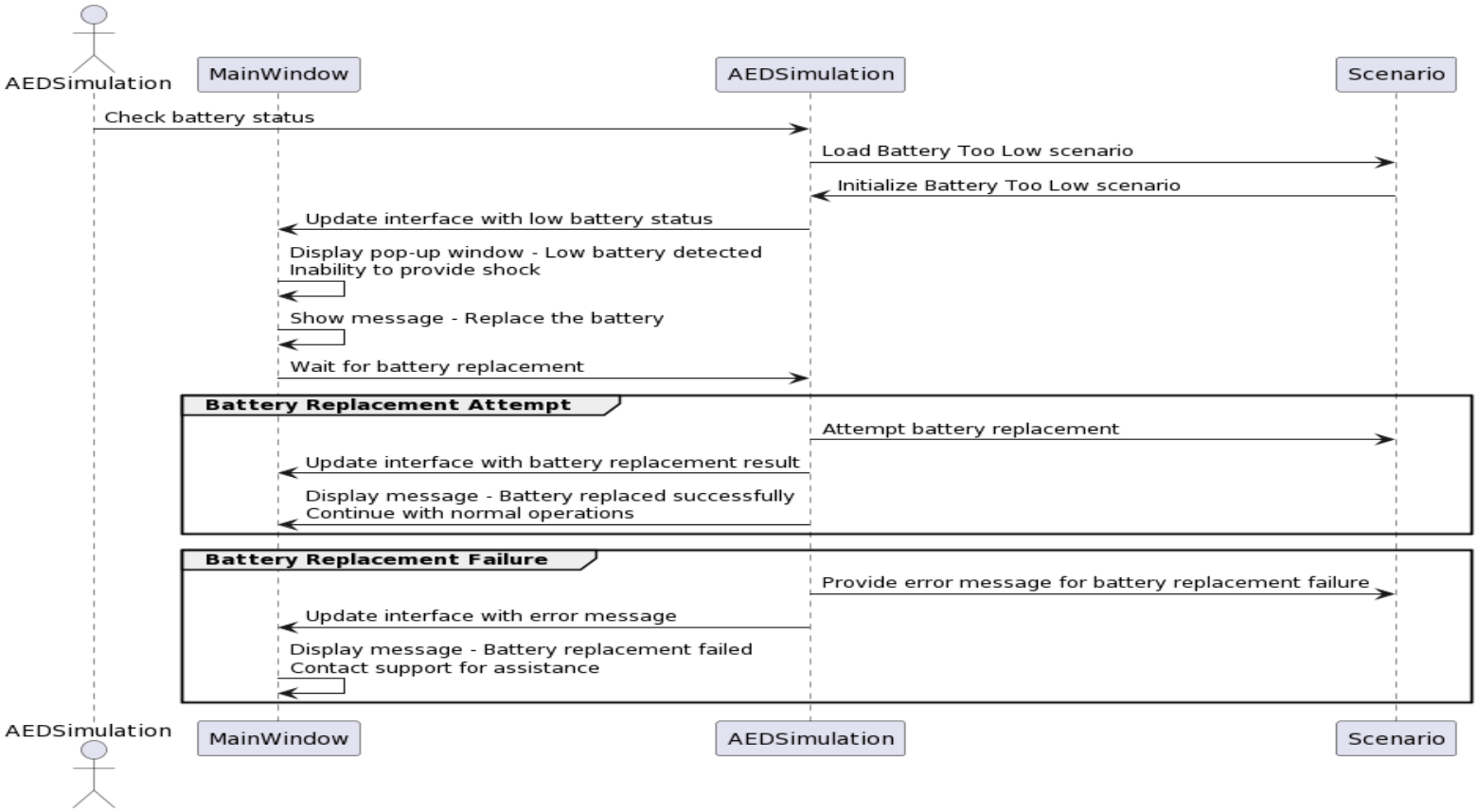


Use case 7: Battery Too Low

1. **Name:** Battery Too Low
2. **Actor:** AED Simulation
3. **Pre-conditions:** AED is turned on with a low battery.
4. **Main Success Scenario:**
    a. AED detects a low battery.
    b. A pop-up window notifies the user about low battery and the inability to provide shock to the patient.

c. AED instructs the user to replace the battery.

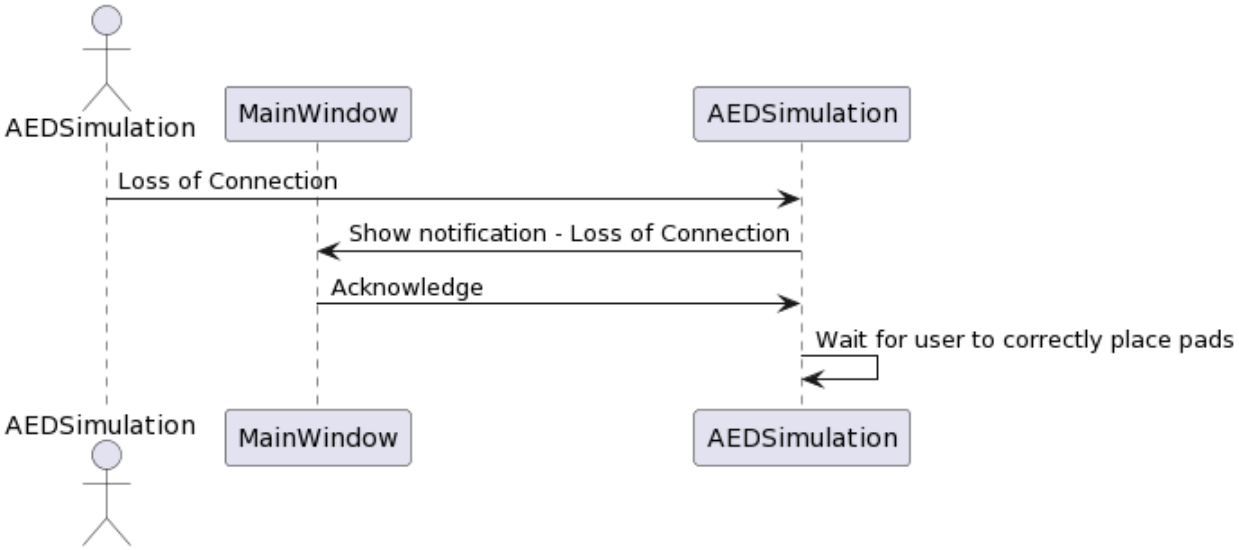
5. **Extensions:**
    a. If the low battery situation still persists despite battery replacement attempts(step 4c), the AED provides an error message.
    b. If there is a loss of connection during the process, it goes back to placing pads.
6. **Success Guaranteed:** AED notifies the user of low battery and instructs them to replace battery for proper functionality.

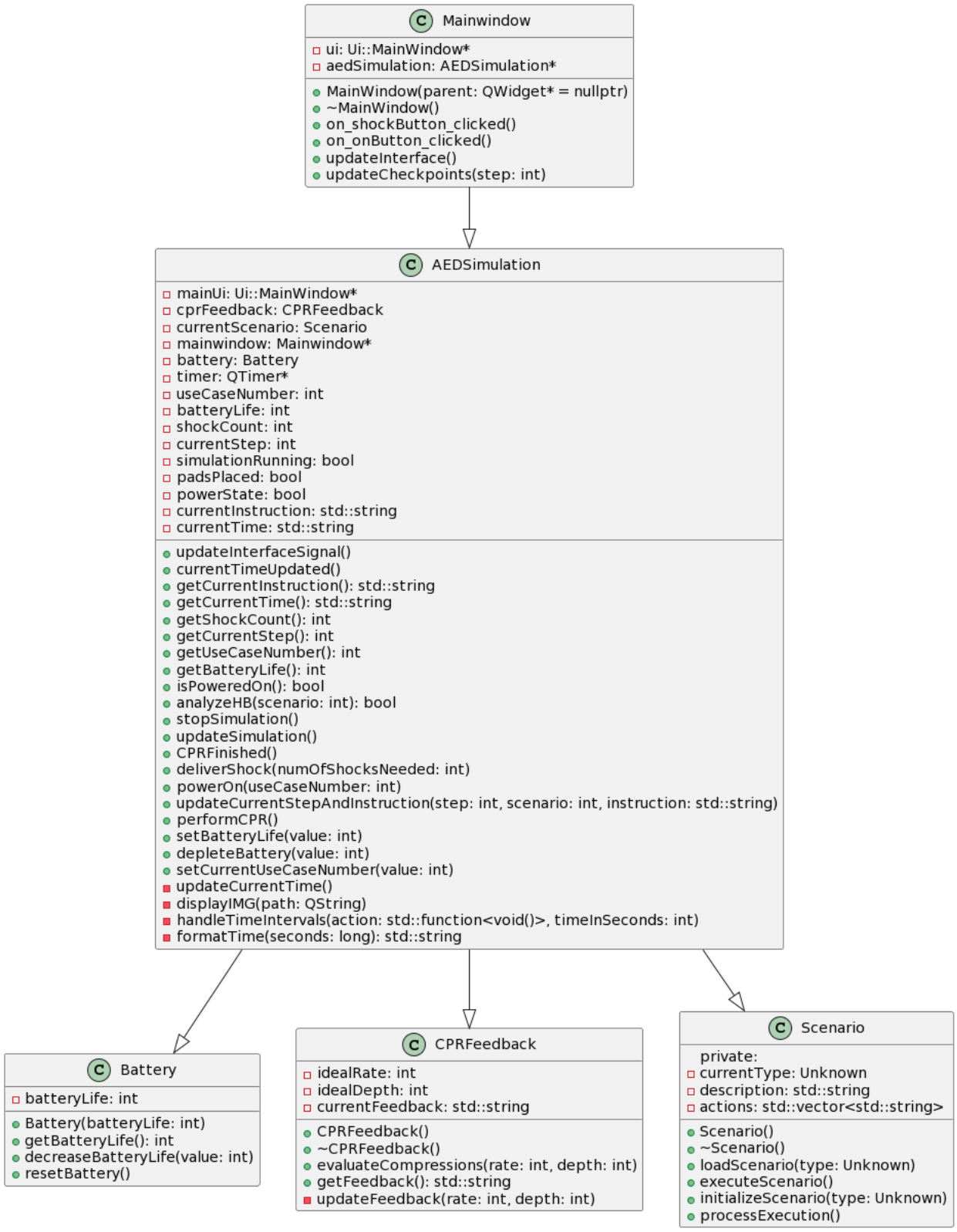


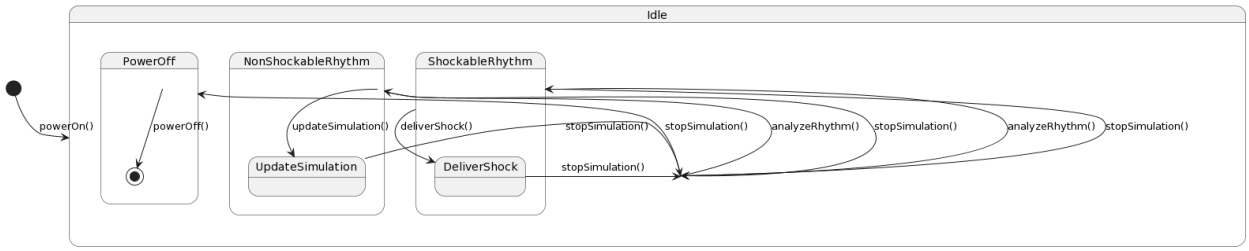
Use Case 8: Loss of Connection

1. **Name:** Loss of Connection
2. **Actor:** AED Simulation
3. **Pre-conditions:** AED is turned on, and electrode pads are connected to the patient.
4. **Main Success Scenario:**

    a. AED detects a loss of connection between the electrode pads and the patient.
    b. A pop-up window notifies the user about the loss of connection.
    c. AED waits for the user to correctly place pads on the patient.
5. **Extensions:** None
6. **Success Guaranteed:** AED recognizes the loss of connection and notifies the user.




Mainwindow
- ui: Ui::MainWindow*
- aedSimulation: AEDSimulation*
+ MainWindow(parent: QWidget* = nullptr)
+ ~MainWindow()
+ on_shockButton_clicked()
+ on_onButton_clicked()
+ updateInterface()
+ updateCheckpoints(step: int)
AEDSimulation
- mainUi: Ui::MainWindow*
- cprFeedback: CPRFeedback
- currentScenario: Scenario
- mainwindow: Mainwindow*
- battery: Battery
- timer: QTimer*
- useCaseNumber: int
- batteryLife: int
- shockCount: int
- currentStep: int
- simulationRunning: bool
- padsPlaced: bool
- powerState: bool
- currentInstruction: std::string
- currentTime: std::string
+ updateInterfaceSignal()
+ currentTimeUpdated()
+ getCurrentInstruction(): std::string
+ getCurrentTime(): std::string
+ getShockCount(): int
+ getCurrentStep(): int
+ getUseCaseNumber(): int
+ getBatteryLife(): int
+ isPoweredOn(): bool
+ analyzeHB(scenario: int): bool
+ stopSimulation()
+ updateSimulation()
+ CPRFinished()
+ deliverShock(numOfShocksNeeded: int)
+ powerOn(useCaseNumber: int)
+ updateCurrentStepAndInstruction(step: int, scenario: int, instruction: std::string)
+ performCPR()
+ setBatteryLife(value: int)
+ depleteBattery(value: int)
+ setCurrentUseCaseNumber(value: int)
- updateCurrentTime()
- displayIMG(path: QString)
- handleTimeIntervals(action: std::function<void()>, timeInSeconds: int)
- formatTime(seconds: long): std::string
Battery
- batteryLife: int
+ Battery(batteryLife: int)
+ getBatteryLife(): int
+ decreaseBatteryLife(value: int)
+ resetBattery()
CPRFeedback
- idealRate: int
- idealDepth: int
- currentFeedback: std::string
+ CPRFeedback()
+ ~CPRFeedback()
+ evaluateCompressions(rate: int, depth: int)
+ getFeedback(): std::string
- updateFeedback(rate: int, depth: int)
Scenario
private:
- currentType: Unknown
- description: std::string
- actions: std::vector<std::string>
+ Scenario()
+ ~Scenario()
+ loadScenario(type: Unknown)
+ executeScenario()
+ initializeScenario(type: Unknown)
+ processExecution()



ID	Requirements	Related Use Case	Fulfilled By	Test	Description
1	Design GUI resembling AED Plus's display	Use Case1- Power on	AEDSimulation.cpp and Scenario.cpp	Interface usability test	Ensure GUI is intuitive and displays necessary info
2	Simulate Cardiac arrhythmia detection	Use case 5- irregular heartbeat (VF) Use case6- irregular heartbeat (VT)	AEDSimulation.cpp and Scenario.cpp	Functional test	Detect VF and Vt rhythms; provide visual/textual prompts
3	Provide real-time CPR feedback	Use Case 3- regular heartbeat Reading. Use case4- Regular Heart(flatlined)	AEDSimulation.cpp and Scenario.cpp	Feedback accuracy test	Measure and assess chest compressions

4	Develop visual prompts for emergency scenarios	Usecase4-Regular Heartbeat(flatlined) Usecase5-Irregular Heartbeat (VF) Usecase6 – irregular heartbeat (VT)	AEDSimulation.cpp, Scenario.cpp and CPRFeedback.cpp	Scenario simulation test	Guide user through rescue situation
5	Interactive system for AED operation simulation	Use case1-power on Usecase2-pad placement Usecase3-Regular heartbeat reading. Usecase4-regular heartbeat(flat lined) Usecase5-irregular heartbeat Use case 6-irregular heartbeat Usecase7-battery too low	AEDSimulation.cpp, Scenario.cpp, Battery.cpp and CPRFeedback.cpp	User interaction test	Allow users to perform actions like in real world AED
6	Create simulated scenarios for software testing	Usecase5-irregular heartbeat (VF) Usecase6-irregualr	AEDSimulation.cpp, Scenario.cpp and CPRFeedback.cpp	Scenario test	Test with various patient conditions

		heartbeat (VT)			
7	Display simulated ECG waveform and device status	Usecase1-power on Usecase2-pad placement Usecase3-regular heartbeat reading. Usecase4-regular heartbeat(flat lined) Usecase5-irregualr heartbeat (VF) Usecase6-irregular heartbeat (VT) Usecase7-battery too low	AEDSimulation.cpp and Battery.cpp	Display functionality test	Accurately simulate ECG and device status
8	Instruction for shock delivery and CPR in emergencies	Usecase4-Regular heartbeat(flat lined) Usecase5-irregualr Use case5-irregular heartbeat heartbeat (VF) Usecase6-irregualr heartbeat (VT0	AEDSimulation.cp, Scenario.cpp and CPRFeedback.cpp	Compliance test	Ensuring instructions are clear and accurate

	Pad Placement Check	Uses cases: 2,3,4,5,6,8.	Scenario.cpp		
9	Compression Depth Analysis	Use cases: 2,3,4,5,6,8	Scenario.cpp, CPRFeedback.cpp	Compliance and Scenario Test	The device is able to give the user feedback on their compressions, feedback on both the rate and depth of the compressions.
10	Device performs a self test	Use cases: 1,7	AEDSimulation.cpp, Scenario.cpp, Battery.cpp	Compliance and Scenario Test	The device should perform a self-test check, and lets the user know whether the battery is too low or if the pads are functional

Explanation of our design decision:

The mainwindow has event-listeners on the "Turn on" Button, when it is pressed it sends a request to the AEDSimulation class but that depends on the use case number selected form the drop down button (1-8)
The scenario class has a list of functions to call depending on the use case selected
When the heartbeat analysis is being conducted, the scenario class then sends a request to the AED Simulation class to perform an analysis
When the CPR is performed, the scenario class then sends a request to the AED Simulation class which then sends a request to the CPRFeedback class

Design Patterns used:

1) **Singleton pattern: Each of our classes have only one instance and they all provide a global access to them**
2) **Observer pattern: When the different scenarios changes, all the other classes are changed accordingly**
3) **Strategy Pattern: There are presence of switch statements which holds the function patterns depending on what the use case number is**