DENAS-PCM6 Simulator	
test case	**test procedure**
power button	User presses the power button, displayed by a red power on symbol. Then the slot function changeToMainPage() gets called in the Display class through the signal/slot connection in Cpu. This displays the main menu to the user if they were on the start page. If they were on the main menu page, then they get sent back to the start page.
selecting a menu option (navigation buttons)	User presses the up and/or down buttons to navigate to a desired menu option. The up and down buttons are represented by a green up arrow and a green down arrow respectively. Both the up and down buttons are connected in the Cpu class. When the up button is clicked, signaled by pushButton()'s clicked() function, the display's class slot function navigateUpLIst() is called, which moves the menu selector up one menu option. When the down button is clicked, signaled by pushButton()'s clicked() function, the display's class slot function navigateDownLIst() is called, which moves the menu selector down one menu option.
programs	User presses the "OK" button on the "Programs" option. Once the OK button is clicked, Display's selectChoice() function gets called by the signal/slot connection in the Cpu class. selectChoice() then calls setCurrentIndex() which displays the Programs page.
starting a program	The user navigates to the desired program with the up/down buttons. Then once the "OK" button is pressed on a desired program, Display's selectChoice() function gets called by the signal/slot connection in the Cpu class. selectChoice() then calls startProgram(int) based on the program number it is passed. This function will create the correct program object (Allergy, Bloating, Trauma, or Kidney), and the set display's therapyPage's mins, secs, name, frequency, power level. The user

	is then displayed the therapyPage which will have the correct program name, frequency, timer, and frequency. The user must then set a power level (with the left and right arrow buttons) and place the electrode on skin (with the electrode checkbox). The user then presses the “start” button on the therapy page, which calls therapyPage’s startTimer() function. This starts the timer on the page.
frequency	The user presses the “OK” button on the “Frequency” option. Once the OK button’s signal function clicked() is called, Display’s slot selectChoice() function gets called by the slot/signal connection in the Cpu class. selectChoice() then calls setCurrentIndex(). The Frequency page is now visible to the user.
setting a frequency	The user uses the frequency slider to set to a desired frequency and will press the start Frequency button when ready. When the slider is changed, the signal function valueChanged(int) is called which calls frequencyPage’s slot function showValueOnDisplay(int). When the user presses the start Frequency button, the signal function startFrequency() then gets called through the slot/signal connection in the Cpu class which sets frequencyTherapyPage’s name, power and frequency, through the object’s setName() and setFrequencyAndPower() function.
Starting a frequency	The user then presses the “start” button on the frequency therapy page, which calls frequenctTherapyPage’s startTimer() function. This starts the timer on the page. Instead of the timer couting down like on the program page, the timer counts up on the frequency page.
adjusting power level	If the user is in the program or frequency page, they must set a power level before starting. The user will then press either the left or right arrow buttons to decrease and increase the power respectively. When the right arrow is pressed, the slot function

	increasePower() get's called by the signal/slot connection in the cpu class and increases the power level by 1. Power level is represented by a QLCDNumber(). PowerLevel's increasePower(power) function first checks to see if the user is on the program page, then increases it's attribute power by 1. It then emits emitPowerLevel() which will update the QLCDNumber's diplay to the window. When the left arrow is pressed, the signal function decreasePower() get's called by the signal/slot connection in the Cpu class decreases the power level by 1. PowerLevel's decreasePower() function first checks to see if the user is on the program page, then decreases it's attribute power by 1. It then emits emitPowerLevel(power) which will update the QLCDNumber's diplay to the window.
battery dies	The battery object inherits the QProgressBar and when its value reaches 0, the emit function emitPowerOff() gets sent out in battery's function drainBattery(), which ends up calling the function rechargeBattery() in the cpu as a slot. This function in battery will increase the value of battery by increments off 25 while sleeping for 1 second inbetween each increase, while also displaying the powerDownPage to the user. In display's rechargeBattery(), the function emitChangeToStart() get's emitted, which will call the function changeToStartPage() in Display once battery's power reaches 100. Once it reaches 100, the startPage is displayed to the user.
recording a therapy	When the user presses the end button on either the program of frequency page, a fnction called endTimer() gets called. The program will prompt the user with a QMessageBox asking if they want to save the treatment session. If yes, then endTimer() initializes a Recording object through the function createRecording(). This function will emit emitRecording(r) with r being the

	initialized recording object. The Cpu object receives this signal and ViewHistoryPage’s addRecording(Recording *) by signal/slot connection. This function converts all of the recording object’s attributes to QStrings and then pushes it to a list, which gets added as a widget on ViewHistoryPage’s layout.
viewing recording(s)	When the user presses the “OK” button on View in the History menu, Display sets the current index of its QStackedWIdget to ViewHistoryPage. Its layout will display all of the contents of its recordings list.
clear history	When the user presses the”OK” button on ViewHistoryPage’s clear option, emitClearHistory is sent out which calls the function clearHistory() in ViewHistoryPage. This function clears the list in itself using list->clear().
no implementation	When the user presses on any menu option that has not been implemented, display set’s the current index to the noImplementationPage.