ID	Requirement	Related Use Case	Implemented By	Tested By	Description
1	The application interface contains buttons, electrodes and images.	N/A	1. **mainwindow.ui** 2. **mainwindow.h** 3. **mainwindow.cpp**	Press the run button in QT creator to observe the ui.	Using QT's built-in user interface framework, the Oasis Pro was replicated. If you click on any button and combobox in the Oasis Pro, an action will occur.
2	The Oasis Pro is powered on through the use of a power button.	Turn Oasis Pro On Use Case 1 (UC1)	1. **mainwindow.ui** 2. **mainwindow.h** 3. **mainwindow.cpp**	Once you start the program, the mainwindow GUI will show up on the screen first. On the right side of the GUI there is a yellow power button. If you click the power button once it will turn on the device and the led above the power button will turn green, however some buttons will be disabled. The light on the bar graph will light up, for the first 3 seconds it will indicate the amount of battery power. If you click the power button a second time, all buttons and icons will be enabled and visible.	The Mainwindow class is used to perform the functions of the power button which include turning on the Oasis Pro. When the power button is clicked once it sends a signal to Mainwindow to enable the Oasis Pro bar graph and all buttons in the Oasis Pro. The on_powerBtn_clicked() function is responsible for keeping track of the number of clicks to the power button. In this case, if the user clicks the power button once then it will turn on the device. But by default, on the first click a variety of icons and buttons will be disabled. The icons for both duration and session, CES graph, and left and right Ear connection will be disabled. Once the user clicks the power button a second time, it will enable all icons and

					buttons. To summarize, the Mainwindow class is responsible for keeping track of the number of times the Power button is pressed through the on_powerBtn_clicked() function that uses if statements to keep track of the number of times the power button is clicked. Then the mainwindow.ui is used to display the GUI and the resulting enablement and disablement of the buttons/icons when the power button is pressed.
3	The Oasis Pro is powered off through the use of a power button.	Turn Oasis Pro Off Use Case 2 (UC2)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Once you start the program, the mainwindow GUI will show up on the screen first. On the right side of the GUI there is a yellow power button. If you click the power button once it will turn on the device. If you click the power button a second time all buttons and icons will be enabled and visible. **However, if you click the power button a third time, it will completely turn off the device.** Furthermore, in the event that no session is selected within 2 minutes, the Oasis Pro will automatically turn off.	The Mainwindow class is used to perform the functions of the power button which include turning off the Oasis Pro. When the power button is clicked a third time it sends a signal to Mainwindow to disable the Oasis Pro bar graph and all buttons in the Oasis Pro. The on_powerBtn_clicked() function is responsible for keeping track of the number of clicks to the power button. In this case, if the user clicks the power button a third time, then it will turn off the device. Thus, all buttons will be disabled and lights turned off. The turnOffNoSessionSelected() function in

					Mainwindow class is used to completely turn off the device if the user does not select a session after 2 minutes. To summarize, the Mainwindow class is responsible for keeping track of the number of times the power button is pressed through the on_powerBtn_clicked() function that uses if statements to keep track of the number of times the power button is clicked. Then the mainwindow.ui is used to display the GUI and the resulting disablement of the buttons/icons when the power button is pressed for the third time.
4	Ending a Session.	Ending a Session Use Case 3 (UC3)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Once you start the program, the mainwindow GUI will show up on the screen first. On the right side of the GUI there is a yellow power button. Click the power button twice and select both a duration and session using the left and right arrows. Then select the checkmark button. Once the connection test ends and the session starts, wait until the session ends by monitoring the duration timer. The session will end with Soft Off, meaning that the graph will scroll from 8 to 1 to confirm that Soft Off is in	The Mainwindow class is used to perform the functions of ending a session which include scrolling the graph from 8 to 1 and turning off the Oasis Pro. The connectionTestMain() function contains an if statement that if the power button is pressed (numberOfTimesPowerBtnClicked == 0) and there is currently a session running (sessionOnOrOff == true) then the graph led's will scroll from 8 to 1 through the descendEndSession() function meaning that SoftOff is currently in

				progress. After Soft Off the device will power off. The same process will occur if the user decides to end a session early by pressing the power button during a session, Soft Off will occur.	progress. Once the led finishes its scrolling animation, the device will turn off through the deviceOff(), iconsOff() and offConnect() functions. To summarize, the Mainwindow class is responsible for keeping track of when a user ends a session early by pressing the power button, then the device will activate SoftOff. Then the mainwindow.ui is used to display the GUI and the resulting Soft Off mode of the scrolling of the graph from 8 to 1 when the power button is pressed during a session.
5	The application battery depletes a function of length of therapy, intensity and connection to skin.	Show Battery Level Use Case (UC4)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Start the treatment and observe the bar graph displaying the battery every 10 seconds. The battery level will start lowering as the treatments continue until the end.	The Mainwindow class keeps a record of the battery level of the Oasis Pro, which the MainWindow ui can then use to update the bar graph. As a therapy is started and continues to run the battery level will deplete as a function of the length of the therapy (duration that is chosen for the session), the intensity that is chosen during the session and whether or not the user selected connected or disconnected in the apply to “Apply to Skin” combo box.

6	Show Battery Level	Show Battery Level Use Case (UC4)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Once you start the program, the mainwindow GUI will show up on the screen first. On the right side of the GUI there is a yellow power button. If you click the power button once, the battery level will be displayed on the bar graph for a couple of seconds when the Oasis Pro is first turned on. Start a session by selecting both duration and session and select the checkmark button. When the session starts, the battery level will be displayed periodically on the bar graph while the session is running. As you use the Oasis Pro more and run sessions, the battery level will start decreasing. When the battery level is getting low, the graph will display two (2) bars and blink. When the battery level is critically low, the graph will display a single (1) blinking bar. If this warning occurs during a session, the session will end early, and the battery indicator will continue to blink until the device turns off. The battery has to be replaced by clicking the "replace battery" button before the device can be used again.	The Mainwindow class is used to perform the functions of showing the battery level. Upon turning on the device by pressing the power button first, the on_powerBtn_clicked() function checks if the power button is clicked once (numberOfTimesPowerBtnClicked == 0) and then displays the battery level in the bar graph by flashing certain led's, through the following, batteryStartTimer -> start(500);. To summarize, the Mainwindow class is responsible for keeping track of when the battery level should be displayed. Then the mainwindow.ui will explicitly show the changes in the bar graph to show the battery level.

7	User selects a session	User selects a session Use Case (UC5)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Once you start the program, the mainwindow GUI will show up on the screen first. On the right side of the GUI there is a yellow power button. Click the power button twice. The standard session group durations will be enabled (20-minute, 45 minutes, and user designated). Also, each group offers 4 sessions that will be enabled underneath (Alpha, SMR, Beta, Theta). Select both a duration and session using the left and right arrows. The session number will be indicated by the lit number in the bar graph. Then press the select button (checkmark) to start a highlighted session. The session number will flash, and the session will begin after a five second delay.	The Mainwindow class is used to perform the functions of selecting a session which include selecting the duration and session type through the use of arrow buttons, which then highlight a session number. This number then flashes before beginning the session after a 5 second delay. The selectionBtn_clicked() function is called when the user presses the selection button (checkmark). It will then disable the selection button and turn off the Qtimer. Then it will store the session and duration values in an array called sessionArray. It will then flash the selected session number for 3 seconds (3000 miliseconds)by using the flashSelectedLevelAfterSelection() function that starts the Qtimer named timerFlashes. Then after flashing the session number, the delay5Seconds() function is used to start the fiveSecondsDelay timer to indicate that the session is about to start. Finally, the continueAfter5Seconds() function is called after the 5 seconds has been finished and then the

					session will start. To summarize, the Mainwindow class and the functions described above are responsible for both duration and session selection and highlighting of the session number in the bar graph. Additionally, they're responsible for starting a session by selecting the select button. Then the mainwindow.ui is used to display the GUI and the resulting duration and session selection outcomes.
8	Starting a Connection Test	Starting a Connection Test Use Case (UC6)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Once you start the program, the Mainwindow GUI will show up on the screen first. On the right side of the GUI there is a yellow power button. Click the power button twice. The standard session group durations will be enabled (20-minute, 45 minutes, and user designated). Also, each group offers 4 sessions that will be enabled underneath (Alpha, SMR, Beta, Theta). On the right side of the GUI, there will be an "Apply to Skin" combo box. Select "Connected" in the "Apply to Skin" combo box.	The Mainwindow class is used to perform the functions of starting a connection test which include selecting the duration and session through the use of arrow buttons, highlighting a session number and flashing the number and beginning the session after a 5 second delay. The connectivity test starts at the beginning of each connection test by entering a test mode. The CES Mode light will then blink. The bar graph will then display the status of the connection. Then the connection will go blank and the intensity may be adjusted as soon as

				Furthermore, in the “Dry/Wet” combo box, select “Wet”. After doing that, select both a duration and session using the left and right arrows. The session number will be indicated by the lit number in the bar graph. Then press the select button (checkmark) to start a highlighted session. The session number will flash, and the session will begin after a five second delay. At the start of each session, the Oasis Pro will check for a connection by entering a test mode. The CES Mode light will blink, and then the graph will display the status of the connection. The bar graph will then display a “no connection” status (blinking red led 7,8), “okay” connection (solid yellow led 6,5,4), or “excellent” connection (solid green led 1,2,3). The Oasis Pro will run with either an Okay or Excellent connection. Once a connection has been confirmed, the display will go blank. Once the connection test ends, the intensity can be adjusted.	the connection test ends. The delay5Seconds() function calls the connection test function after the five seconds delay has ended. Furthermore, multiple icons will change colors to indicate connectivity such as the graphSessionOn() function that turns the CES graph on and sets the corresponding Boolean variable to true. Furthermore, both the left and right ear icon will light up if the “Apply to Skin” combo box is set to be connected. The function blinkCounter() is called when the QTimer named timerBlinked ever enters a timeout. Using if statements, we are able to blink led 7 and 8 if the “Apply to Skin” combo box is disconnected. Furthermore, we are also able to blink led’s 4,5, and 6 if both the “Apply to Skin” combo box is connected and the “Dry/Wet” combo box is set to “Dry”. Lastly, we can blink led’s 1,2 and 3 if both the “Apply to Skin” combo box is connected and the “Dry/Wet” combo box is set to “Wet”.

					To summarize, the Mainwindow class and the functions described above are responsible for the connectivity test and blinking leds based on either bad, okay or excellent connection in the bar graph. Then the mainwindow.ui is used to display the GUI and the resulting connectivity test outcomes.
9	Treatment displays the time of treatment	N/A	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h** 4. **TherapyRecord** 5. **Database**	Select a duration and session and then click the select button. The session will start, and you will see the allocated time on the far right of the GUI.	The Mainwindow class contains a Qtimer attribute, which maintains the timer of how long the therapy has been running since the session started and connectivity test ended. The Mainwindow.ui class will display the countdown of the treatment durations when connectivity ends and the session has started. The Database class will save the TherapyRecord when the session ends
10	Treatment time advanced only when electrode connected to the skin. (Reference to requirement 9)	Activate Electrodes Use Case (UC9)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h** 4. **TherapyRecord** 5. **Database**	Select either connect or disconnect in the “Apply to Skin” combo box. Start a session treatment by clicking the select checkmark button. Once the connectivity test ends, observe the timer. It would start counting down if it’s connected to the skin. However, if it's disconnected, the timer will not countdown.	The Mainwindow class will check if the device is applied to the skin through the “Apply to Skin” combo box. The user can change the “Apply to Skin” combo box to either connect or disconnect from skin. However, the therapy timer will only start counting down if the “Apply to Skin” combo

					box is set to connect.
11	Treatment time does not advance or pauses when electrodes are not applied to skin. (Reference to requirement 9)	Activate Electrodes Use Case (UC9)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h** 4. **TherapyRecord** 5. **Database**	Select either connect or disconnect in the "Apply to Skin" combo box. Start a session treatment by clicking the select checkmark button. Once the connectivity test ends observe the timer. It would start counting down if it's connected to the skin. However, if it's disconnected, the timer will not countdown.	The Mainwindow class will check if the device is applied to the skin through the "Apply to Skin" combo box. In this case, the therapy timer will not start if the "Apply to Skin" combo box is set to disconnect.
12	The Oasis Pro supports 3 durations.	User selects a session Use Case (UC5)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Click the Power Button two times to enable the durations menu. Once enabled, use the left and right arrow buttons next to the durations to highlight/select an appropriate session duration.	The user uses the duration left and right (functions are on_durationLeft_onclick() and on_durationRight_onclick()) buttons to select one duration (either 20 seconds, 45 seconds, or a custom time – this custom time is a time inputted by the user in seconds). By default, 20 seconds is chosen. This time value (either 20 seconds, 45 seconds, or custom time) is stored in the newRowItemDuration variable. This value is based on the position or index of the selected duration item in the "listDuration" widget. This means that if the user selects 20 seconds, the value stored in that variable is 0 (position of the item in the list), 45 seconds would be 1, and custom time would be 2.

					If custom time is chosen, the time inputted by the user also gets stored in the “customDuration” variable. After selecting their duration, the newRowItemDuration will be used to record their chosen time (“0” represents 20 seconds, “1” represents 45 seconds, and “2” represents custom duration). Based on the value of newRowItemDuration, the actual duration of the session will be stored in the session array at index 0. So 20 will be stored in the array when newRowItemDuration is 0, and 45 will be stored in the array when newRowItemDuration is 1. When newRowItemDuration has a value of 2 (meaning custom time), the “customDuration“ will be stored instead (this only happens once the user presses the select button, which should happen after they also selected a session).

13	The Oasis Pro supports 4 sessions.	User selects a session Use Case (UC5	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Click the Power Button two times to enable the sessions menu. Once enabled, use the left and right arrow buttons next to the sessions to highlight/select an appropriate session.	The user uses the session left and right (functions are on_sessionLeft_onclick() and on_sessionRight_onclick()) buttons to select one session from 4 different sessions (alpha, SMR, beta, and theta). By default, alpha is chosen. The selected session's value is stored in the variable newRowItemSession. A newRowItemSession value of 0 means alpha, value of 1 means SMR, value of 2 means beta, and value of 3 means theta. Once they have chosen their session, the newRowItemSession value will be stored in the session array at index 1 (this only happens once the user presses the select button, which should happen after they also selected a duration).
14	A saved record therapy will save the session duration, session type and last chosen intensity before the session ends.	Record a Therapy Use Case (UC10)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h** 4. **TherapyRecord** 5. **Database**	Once the session ends, the therapy will be automatically recorded into the database. If the session already exists within the database, it will not be recorded.	The TherapyRecord has attributes for session, duration and last chosen intensity. The Database class stores the recorded therapy which will include all three attributes mentioned above. Then the Mainwindow will display those therapy records under the treatment

					history list box.
15	The Oasis Pro will stop working and everything will be disabled (turn white) when the battery reaches 0.	Show Battery Level Use Case (UC4)	1. **mainwindow.ui** 2. **mainwindow.cpp**	In order to replicate this, you must start a session treatment and observe the battery being depleted either through the bar graph that will show every 10 seconds (bar will start dropping) or through Qdebug which will show the battery depleting as well. Once the battery level reaches 0, the Oasis Pro will shut off and all buttons will be disabled. If you try clicking any button it won't work. In order to replenish the battery of the Oasis Pro, you must click the New Battery button.	The Mainwindow class will be alerted when the depletion of the battery has reached 0. The Oasis Pro will not turn on even if the power button is pressed. One must click the New Battery button in order to replenish the battery to 100.
16	At the end of the connection test, the timer starts counting down according to the session duration chosen.	N/A	1. **mainwindow.ui** 2. **mainwindow.cpp**	Start a session by selecting the select button and then wait until the connection test ends. Once the connection test ends, the timer will start counting down.	The Mainwindow class maintains a Qtimer attribute that counts down starting from the session duration that was initially chosen before starting the session. Once the timer reaches 0, the session ends.
17	When using the Oasis Pro, one can choose any of the four users in the user's combo box to record a therapy and add to treatment history.	Record a Therapy Use Case (UC10)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h** 4. **TherapyRecord** 5. **Database** 6. **Administrator** 7. **Guest**	Start a session by clicking the select button and then wait for the session timer to end. Once the session ends (as indicated by the session timer), then the therapy will be automatically added to the user's treatment history.	The therapies are recorded automatically when the session treatment ends (as indicated by the finished countdown of the session timer). The information that is recorded (duration, session last selected intensity) is stored and

					saved to an external database.
18	A user can view their treatment history.	View a saved Therapy Use Case (UC11)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h** 4. **TherapyRecord** 5. **Database** 6. **Administrator** 7. **Guest**	Press the power button twice to turn the device on. From the treatment history list box, select the refresh button to view all recorded treatments for the user chosen in the “User” combo box.	Next to the Oasis Pro GUI, there is a menu Treatment history interface, the user can then click the Refresh button to see a scrollable list of all therapies associated with that specific user. These treatment records are loaded from the database via the Database class into Mainwindow.
19	Adjusting Intensity	Adjusting Intensity Use Case (UC7)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Once you start the program, the mainwindow GUI will show up on the screen first. On the right side of the GUI there is a yellow power button. Click the power button twice. The standard session group durations will be enabled (20-minute, 45 minutes, and user designated). Also, each group offers 4 sessions that will be enabled underneath (Alpha, SMR, Beta, Theta). On the right side of the GUI, there will be an“Apply to Skin” combo box. Select “Connected” in the combo box. Furthermore, in the “Dry/Wet” combo box select “Wet”. After doing that, select both a duration and session using the left and right arrows. The session	The Mainwindow class is used to perform the functions of adjusting the intensity which include increasing and decreasing the intensity of the stimulus. Users will be able to adjust the intensity once the connection test ends. The blinkCounter() function performs a series of functions after finishing all the connectivity tests based on whether the device is connected or not. For this use case, the intensity buttons will be enabled once the connectivity test ends. To summarize, the Mainwindow class and the functions described above are responsible for enabling adjusting the intensity when the connection test ends. Then the mainwindow.ui is used to display the GUI and

				number will be indicated by the lit number in the bar graph. Then press the select button (checkmark) to start a highlighted session. Once the connectivity test ends and the connection has been confirmed, the intensity can be adjusted. To increase the intensity of the stimulus, press the INT up button next to the bar graph. To decrease the intensity, press the INT down button. As you adjust the intensity using the buttons, the intensity number will be highlighted in the bar graph.	the resulting adjustments of the intensity level.
20	Replacing Battery	Replacing Battery Use Case (UC8)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	When the battery level is critically low due to extensive usage of the Oasis Pro, the graph will display a single blinking bar in the bar graph. The user will also get a message in the console stating, “Device has shut down due to critical battery level”. The Oasis Pro will then Power Off and users will be unable to power on the device. In order to power on the device users must click the “New Battery” button in the Replace Battery section. Once this is clicked the battery will replenish itself to 100%.	The Mainwindow class is used to perform the functions of replacing the battery. Users will be able to replenish the battery to 100% if the battery is depleted. The function showBatteryLevel() determines which of the LED’s will blink if a battery level is within a certain level. If the battery is critically low (less than 12) then only one led will blink. Furthermore, the on_newBattery_clicked() function will replace the battery and reset the batteryLevel variable to 100.

					To summarize, the Mainwindow class and the functions described above are responsible for replenishing the battery to 100% when the replace battery button is clicked. Then the mainwindow.ui is used to display the GUI and the resulting replacement of the battery level.
21	Activate Electrodes	Activate Electrodes Use Case (UC9)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h**	Once you start the program, the mainwindow GUI will show up on the screen first. On the right side of the GUI there is a yellow power button. Click the power button twice. The standard session group durations will be enabled (20-minute, 45 minutes, and user designated). Also, each group offers 4 sessions that will be enabled underneath (Alpha, SMR, Beta, Theta). On the right side of the GUI, there will be an “Apply to Skin” combo box. Select Connected in the combo box. Both the left and right ear icon will light up. When a connection test starts, the bar graph will display the connected status of the connection test. Once the connection test ends, the electrode informs the device to advance the time of therapy.	The Mainwindow class is used to perform the functions of connecting/activating the electrodes. The function connectionTestMain() checks the connection by ensuring that the electrodes are activated meaning the “Apply to Skin” combo box is set to be connected. To summarize, the Mainwindow class and the functions described above are responsible for checking the activation of the electrodes when the “Apply to Skin” combo box is set to be connected. Then the mainwindow.ui is used to display the GUI and the resulting electrode activation.

22	Replay a Saved Therapy	Replay a saved Therapy Use Case (UC12)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h** 4. **TherapyRecord** 5. **Database** 6. **Administrator** 7. **Guest**	Once you start the program you will be able to press the refresh button to see a list of all recorded therapy sessions for a selected user. Use the “Up/Down” buttons to navigate the various recordings until the desired one is reached. Press the “Select” button after highlighting the desired recording. This will change the settings of the device to match the recording. These settings include the displayed duration of the session along with the displayed session type and number. From here you can press the select session(Checkmark) button to start the recording. The OasisPro device will execute in the same fashion as when starting a brand new session. Once the recorded session has finished, it will not be saved a second time to avoid having duplicate recordings. If the battery reaches a critical level mid session, then it will not save the session to the database. If the application is not saving sessions then delete the patient.db file from build-Oasis-Desktop-Debug.	The MainWindow, TherapyRecording and Database classes are the main ones used to handle the functionality of the replay feature. The treatmentRefreshBtn_clicked() function is used to load the recordings for the selected user. The treatmentUpBtn_clicked() function is used to go up the list of recordings. The treatmentDownBtn_clicked() function is used to go down the list of recordings. The treatmentSelectBtn_clicked() function is used to match the settings to the device to that of the highlighted recording. The selectionBtn_clicked() function is used to begin the selected recording. The Database and TherapyRecord classes work together in order to record and store the user’s sessions. Refer to 22:Record a Therapy for more information. The MainWindow class is where the bulk of the functionality of this feature is implemented. Once a particular

					recording has been selected, those settings are remembered in various different variables. Those variables are then used when a recorded session has been started via the select session button. The GUI itself then functions the same way as if you were to run an entirely new session.
23	Clear All Therapy History Records	Record a Therapy Use Case (UC10)	1. **mainwindow.ui** 2. **mainwindow.cpp** 3. **mainwindow.h** 4. **TherapyRecord** 5. **Database** 6. **Users** 7. **Administrator** 8. **Guest**	Once you fully turn on the device and replay a therapy record, you can use the “Clear All” button to send a request from the client (GUI) to the SQL Server (Database “patient.db”). The server will return a result of an identity authentication. If the selected user from the combo box is an administrator, the program will ask the user if the operation should be continued by displaying a popup message. If the user clicked the “Yes” button, all therapy history records would be removed from the server, and each item in the list on the client will be cleared. Otherwise, the operation will be cancelled. If the selected user is not an administrator (i.e., a guest), the request would be rejected and a notification will show up.	A user identity is classified as Administrator and Guest based on Users class. Factory method (a creational design pattern) is applied in this project. Each type of user identity has its own privilege. Both administrators and guests can create and replay their therapy records, but only the administrators can delete all therapy history records in the database. On client side (Mainwindow class), the “Clear All” button is used for sending a request to delete all therapy history records in the database. On server side (Database class), the function deleteTherapyHistoryRecords(Users *) uses a user identiy pointer for identity authentication, and removes all

					records from “treatmentHistory” table in the connected database only when the selected user is recognized as an administrator.