COMP3004A Winter 2020 Final Project For Vojislav Radonjic

By Khushal Singh (101094697)
Tejal Darpan (101063975)
Jaewon Cha (101082608)
Shahbaz Dhanju (101088555)

1. Background

The Microcurrent Biofeedback (MB) device is non-invasive medical instrument used by physicians, dentists, therapists, psychologists and chiropractors for the management of acute, chronic, and post-operative pain. It simulates the body's own bioelectrical system, promoting natural healing and reliefs pain of various symptoms. The technology relies on producing micro current output addressed directly to specific type of nerve ending which formulates neuro transmitter chemicals treating both pain and medical conditions.

2. Description

The device has following electronic components:

1- Microprocessor: modulates the electrical signals via electrodes exposed directly through the skin to communicate with the peripheral nervous system for the purpose of therapeutic intervention.
2- Treatment Mode: predefined programs which are sets of microcurrent frequency and specific time. Each program is conducted for specific medical condition.
3- Power Level Keys: increases and decreases the frequency.
4- Display: LCD or LED screen to present user graphical interface and displaying instructions.
5- Speaker: announce specific beeps as notification device
6- Light LED: showing informative light like error and success codes
7- Electrode: consist of:
   a. Output circuitry which sends modulated microcurrent pulses though out the patient's skin.
   b. Pulse Generator.
   c. Detector / Receptor which detect nerve ending when conducted to patients' skin.
8- USB Interface: connects external electrode for treatments that can't be done by the in-housed electrode.

3. Objective

The main objective of microcurrent biofeedback system is it provide a platform for patient and the device operator that allows:

1- Utilize predefined treatment programs to promote healing and relief of certain medical conditions.
2- The ability to adjust the frequency and time interval of the microcurrent signals to tune up the treatment as required.

4. Core Services

The core services on MB devices to the key success are:

1- Simple, easy-to-use built-in functions and self-intuitive user interface
2- Built-in predefined treatment programs.
3- Configuration adjustment capability using basic touch pad interface.
4- Visual and sounding success and error notifications.

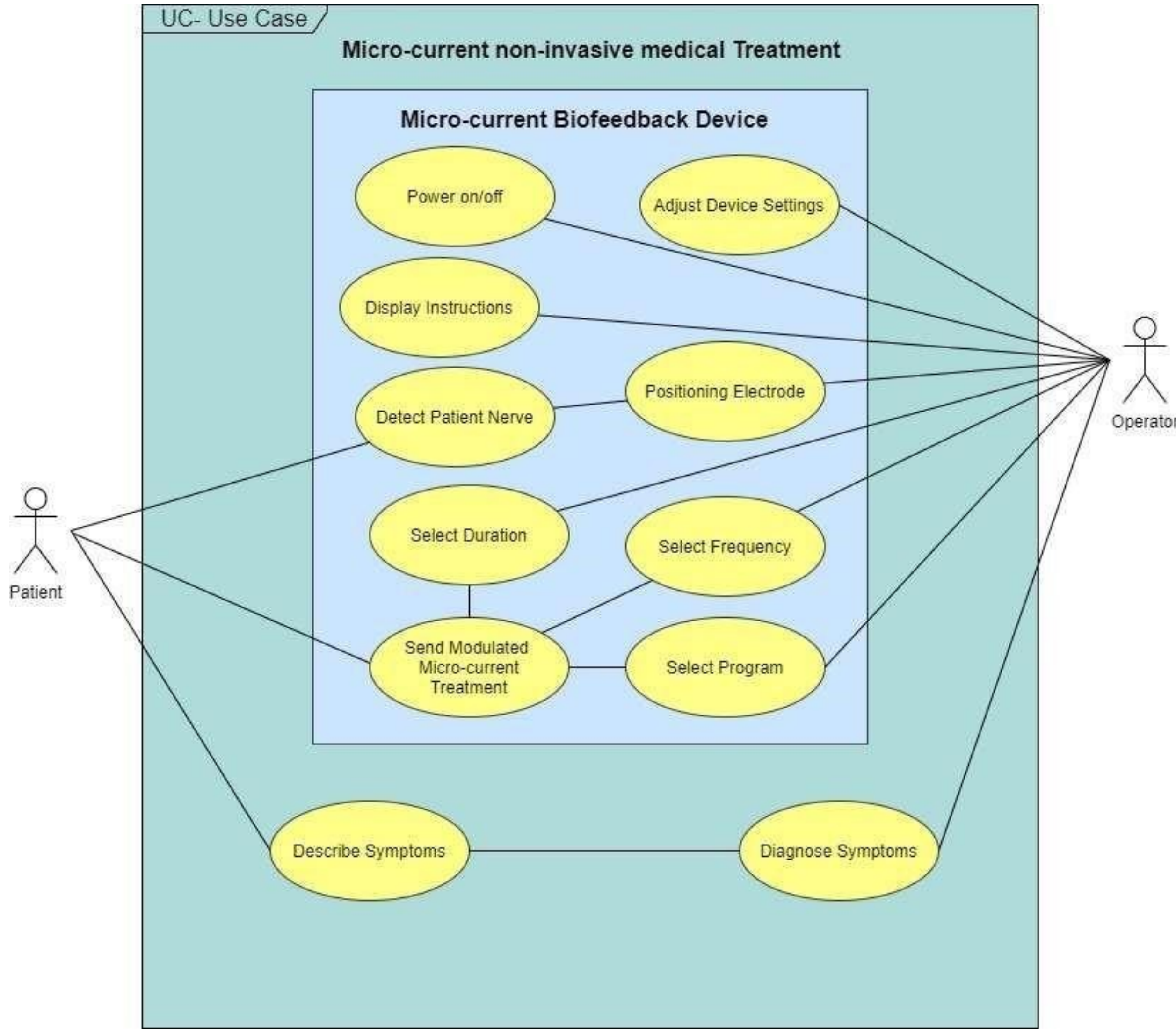
5. Stakeholders

There are two major stakeholders for the MB device:

1- Patient: who seek treatment for a variety of health issues, including concussion, chronic pain, Allergy, neuro-circulatory dystonia, trauma, asthma, nausea, hypertension, diarrhea, arthritis, sports injuries, ligament injuries, muscular strains, migraines, edema and neuropathy,
2- Operator: who are physicians, dentists, therapists, psychologists, chiropractors and in some cases if the patient is trained for home user situation.
3- The patient provides the operator with symptoms and treatment program. This is either the patient is being referred by a license medical service or the operator is licensed to conduct medical diagnostics on the patient and conduct a treatment to the patient. In both cases, these formalities are not designed as part the device functionalities.

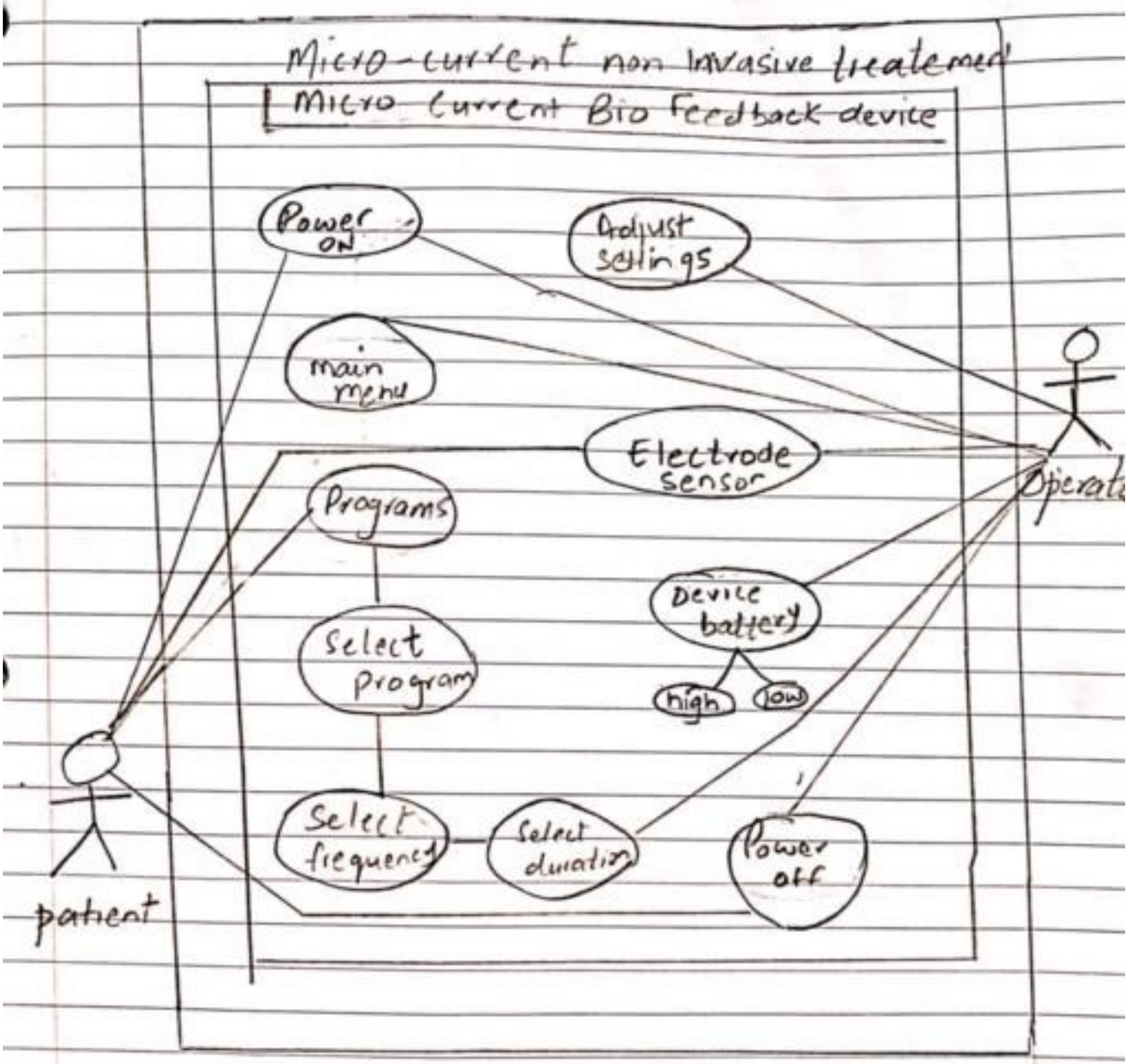
6. Functionalities

The following case model represent main functionalities that the MB device can perform:

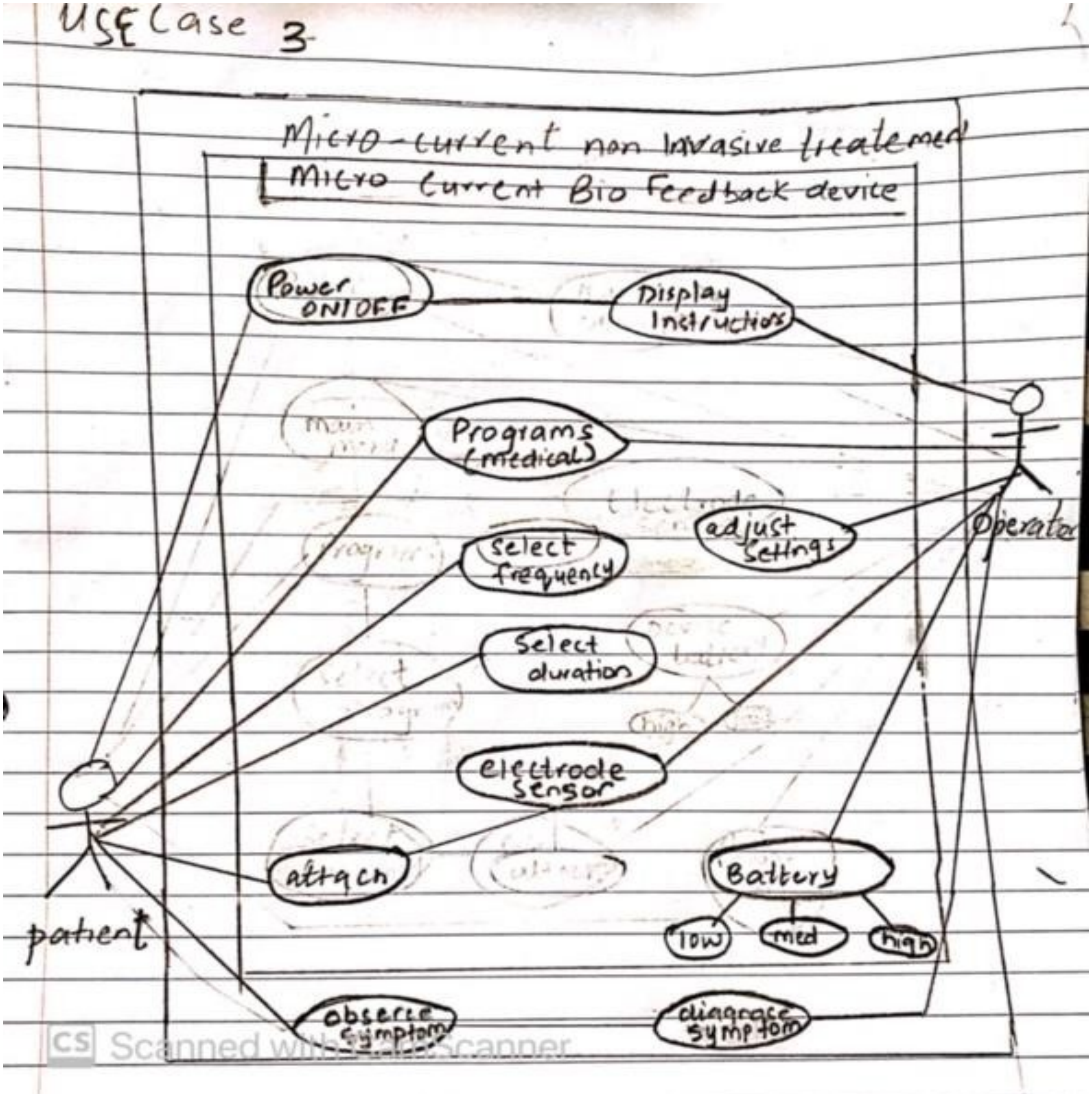


MBD Case Model

USE Case 2



Micro-current non invasive treatement
Micro current Bio Feedback device
Power ON
Adjust settings
main menu
Electrode sensor
Programs
Device battery
Select program
high
low
Select frequency
Select duration
Power off
patient
Operat



1- Select Program: operator can select one of predefined treatment program built-in the device. Each program corresponds to specific medical conditions. In essence, the program is combination of modulated microcurrent frequency and conduction time.
2- Select Duration (MED): the device allows the operator to manually adjust the conduction time using touch pad keys.
3- Select Frequency: the device allows the operator to manually adjust the microcurrent frequency matching the nerve ending signal that is responsible for healing and pain relief. Both the manual duration and

frequency functions can be used for manual calibration as required.

4- Send Modulated Microcurrent: the device processor uses the treatment program or the manually adjusted frequency and conduct time to send the Modulated Microcurrent through the device built-in or attached electrode.

5- Detect Patient Nerve: The device's electrode has three states that is changing based on the signal received from its detectors / receptors.
    a. Not-On-Tissue State: the detector is in idle mode where signaling at 204 volts
    b. First-Contact State: the detector immediately detects when the electrodes are first placed on reactive tissue signaling at 114 volts.
    c. Treatment State: the device continues treatment reaching optimum electrical frequency as set on the selected treatment program or manually adjusted by the operator.
6- Positioning Electrode: Based on the state received from the Detect Patient Nerve function, the processor uses the device' clock to modulate the frequency and duration time intervals and starts the treatment process as per the selected plan or the manual adjust set by the operator.
7- Adjust Device Settings: the device allows the operator to change the device configuration, like display color theme, sound volume, display brightness, clock setting (date and time), alarm clock, economy (power saving) and changing the display language. Additionally, the device allows the operator to change the frequency mapping in case the treatment is conducted on a child as follow:
    a. Disable.
    b. up to one year old.
    c. one to three years old.
    d. four to seven years old.
    e. Seven to twelve years old.
8- Display Instructions: The display unit will present to the operator graphical instructions to guide the operator where to position the electrode for best treatment results. This is a content of images and text describing the human body peripherals where the operator needs to place the device electrode on.
9- Power On/Off: this is to switch on/off the device, as required.

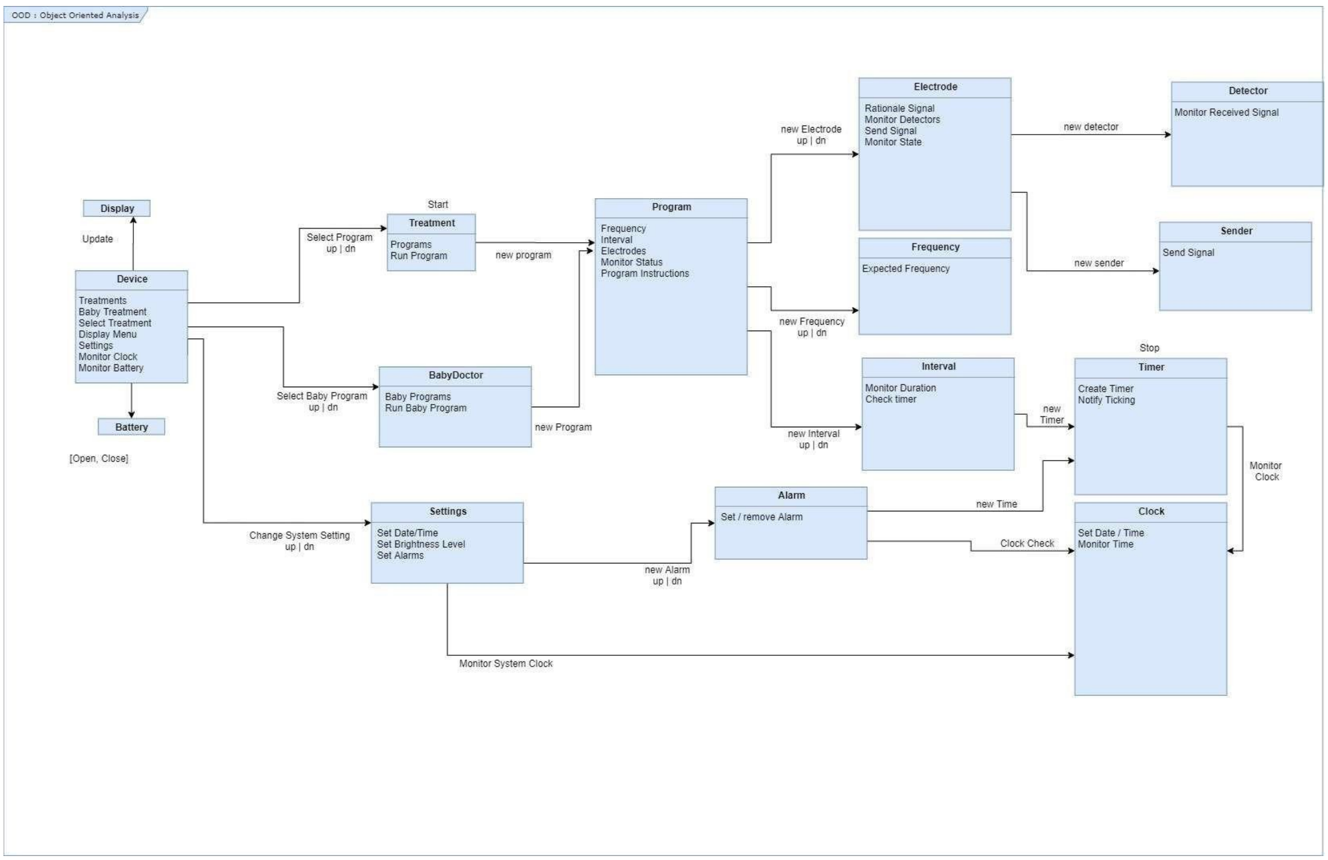
7. Object-Oriented Design (OOD)

The OO Design diagram below shows the object orient class design from high level view. The diagram shows classes that are discovered during the analysis phase as follow:

- Device: this is the main class for the application; it is responsible for initializing the graphical user interface and displaying user interactive menu to select functions offer by the devices. In addition, it keeps the user informed with task progressing and device hardware status like battery level and sending sounds beeps as required.
- Treatment: this class is the major functionality of the device. It is the main driver for treatment offerings and the outcome of Research and Development (RD) of this technology. The operator person uses the navigating keys on the device to select one of the appropriate treatment programs. This class is key to start and initializing any selected program.
- Baby Doctor: this a special treatment class. It factorizes the pre-defined programs to specific ratio based on age of the baby patient. For instance, if the Baby Doctor is set to one-year old patient, the treatment class adjust the selected program to a specific frequency for the signal detected from the patient’s nerve and for the signal sent through the patient’s skin. This additional advantage of the RD offered by the device.
- Programs: this class is pre-defined combination of signal frequency and time interval to be conducted on the patient’s nerve. It contains electrode object which is responsible for detecting patient’s nerve and sending signals. This class is responsible for displaying treatment instructions to the operator user for optimal treatment result.
- Electrode: this is a key class initiated from the program class. It is responsible for detecting the patient’s nerve and sending modulated microcurrent signal. The class keeps monitor of the detectors state and report back to the Program class if the device is ready to send treatment signal.
- Detector: is the class responsible for reading the volt resistance from the patient’s body and report back the state of the device.

- Sender: is the class responsible for sending modulated microcurrent as predefined by the selected program.
- Interval: is duration class that keeps the electrode sending treatment signals within predefined timeframe assigned by the selected program. The class initiates a timer object tracking the duration time using the system clock.
- Timer: is a utility class used by various class offering time intervals that can be utilized in specific functionality. In addition to the treatment duration mentioned above, this class is used to set alarm intervals for user usability features. Also, the main function of this utility class is to send stop state to the program which in return stops the electrode sending treatment signals. Additionally, its ticking monitor, keeps the display screen updated with notifications like timers count down, system data and time, etc.
- Setting: this is a utility class mainly responsible for adjusting the device usability feature like changing the system time and date, setting alarms, changing the display screen brightness level.

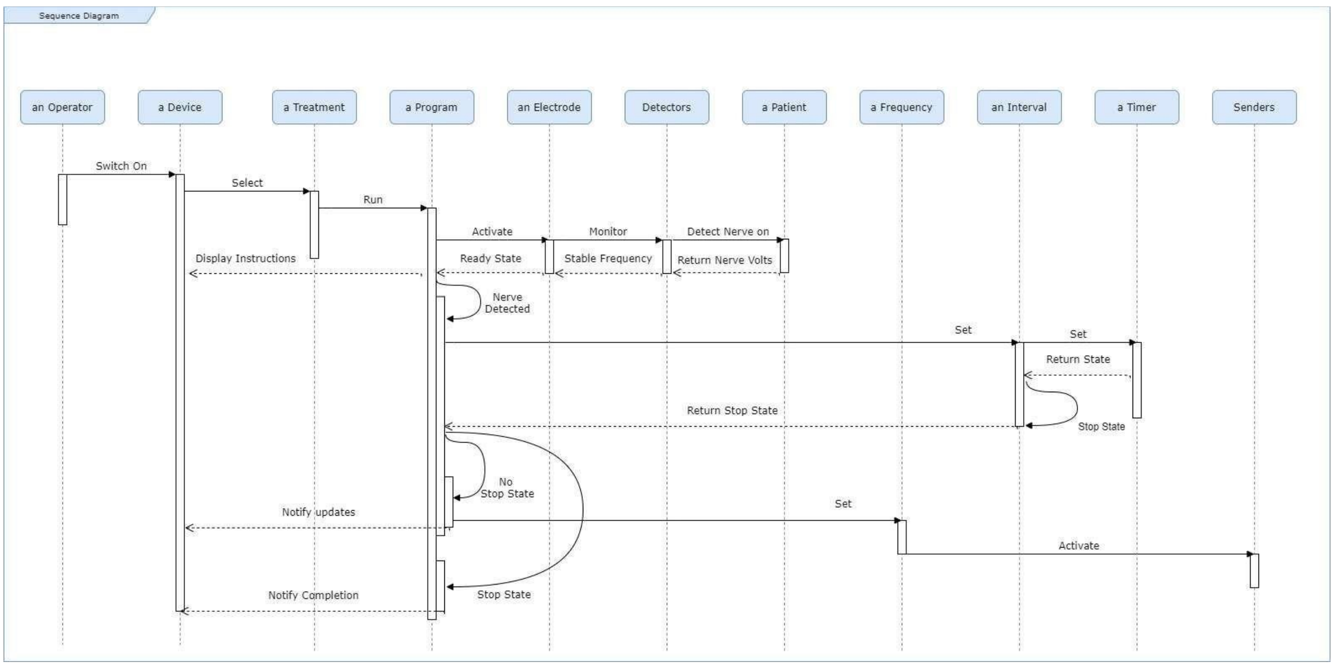
The OOD diagram below shows the user interaction with each class using touch pad keys (power, up, down, left, right, select and back).


OOD : Object Oriented Analysis
Display
Update
Device
Treatments
Baby Treatment
Select Treatment
Display Menu
Settings
Monitor Clock
Monitor Battery
Battery
[Open, Close]
Start
Treatment
Programs
Run Program
Select Program
up | dn
new program
BabyDoctor
Baby Programs
Run Baby Program
Select Baby Program
up | dn
new Program
Program
Frequency
Interval
Electrodes
Monitor Status
Program Instructions
new Electrode
up | dn
Electrode
Rationale Signal
Monitor Detectors
Send Signal
Monitor State
new detector
Detector
Monitor Received Signal
new sender
Sender
Send Signal
Frequency
Expected Frequency
new Frequency
up | dn
Interval
Monitor Duration
Check timer
new Interval
up | dn
new
Timer
Stop
Timer
Create Timer
Notify Ticking
Monitor
Clock
Settings
Set Date/Time
Set Brightness Level
Set Alarms
Change System Setting
up | dn
new Alarm
up | dn
Alarm
Set / remove Alarm
new Time
Clock Check
Clock
Set Date / Time
Monitor Time
Monitor System Clock

8. Sequent Diagram

The sequence diagram below describes one of the major tasks when using the devices.

1. Operator as first step to switch on the device using the power key. The system boots up and load the system systems.
2. The Device use the system setting and identify if the operator adjusts the treatment for Baby Doctor or not, then treatment programs are populated, then finally the device displays a user-friendly menu waiting for a user selection.
3. Once the treatment programs are populated in the menu, the Operator select a program. This initialize an Electrode object and adjusting the treatment signal frequency and the treatment durations. The program does not start till it receives a ready state reported from the detectors
4. Up on initializing the program object, Detector objects are initialized, reading patient nerve signal. Once the detector state reporting READY state, it returns this state to the program which in return initialize the Sending object and start sending treatment signals.
5. Up on initializing the Program object, it initializes an Interval object which sets the treatment duration to the predefined time associate to the program.
6. Once the Interval object is initialized, it creates a Timer object setting the setting the start time to current time and end time to current time plus duration in seconds. The Timer keep triggering a Ticking method which updates the time spent within target duration.
7. The Program object keep monitoring the interval state and its Timer, if it reports Running state, it keeps the Electrode sending the treatment signal till the Timer state reports Stop state which at this point instructs the Electrode to stop sending treatment signal.
8. Within all steps above each step report informational text to the device's display informing the Operator with progressing updates.



Sequence Diagram
an Operator
a Device
a Treatment
a Program
an Electrode
Detectors
a Patient
a Frequency
an Interval
a Timer
Senders
Switch On
Select
Run
Activate
Monitor
Detect Nerve on
Display Instructions
Ready State
Stable Frequency
Return Nerve Volts
Nerve
Detected
Set
Set
Return State
Return Stop State
Stop State
No
Stop State
Set
Notify updates
Activate
Notify Completion
Stop State

9. Activity Diagram

The diagram below illustrates the flow of control in the systems and shows steps involved in the execution of the treatment use case. The diagram below also depicts visual sequence in which objects are initialized and interact to each other.

1- The initial state starts with the device being switched on by the operator pressing on power key. Consequently, the system boots up loading system settings then it displays a menu and waiting for operator's selection.
2- The device checks If the battery is low, then it displays a warning message that the device need to be recharges and can't perform any treatment then goes back and displays the main menu.
3- If the battery is in acceptable level, the device object validates the system setting which has the treatment mode. If the setting is adjusted for Baby Doctor specific age, the treatment object is loaded and adjusting the programs objects with associates Baby patient frequencies and intervals, otherwise, the programs are loaded with default predefined frequencies and time intervals. Finally, the display menu is shown based on the system settings loaded.
4- The Operator selects a treatment program which initializes a new Program object and fires up the run method. Finally, it displays treatment instructions on the device's display. The Operator can exit this information by pressing back key which automatically start detecting patient nerve.
5- Upon initializing the Program object, it initializes Electrode object and Interval object. Both objects feedback state updates to the Program object.
6- The Electrode object initializes Detector objects used as sensors to detect patient nerve signals.
7- The Interval object initialized by the Program initializes a Timer object starts but does not count down the treatment duration until the program gives that start count down command.
8- Meanwhile, the Detector object sense the patient nerve and once it reaches the expected volts as predefined by the program, it sends back state update to the program object informing that the device is

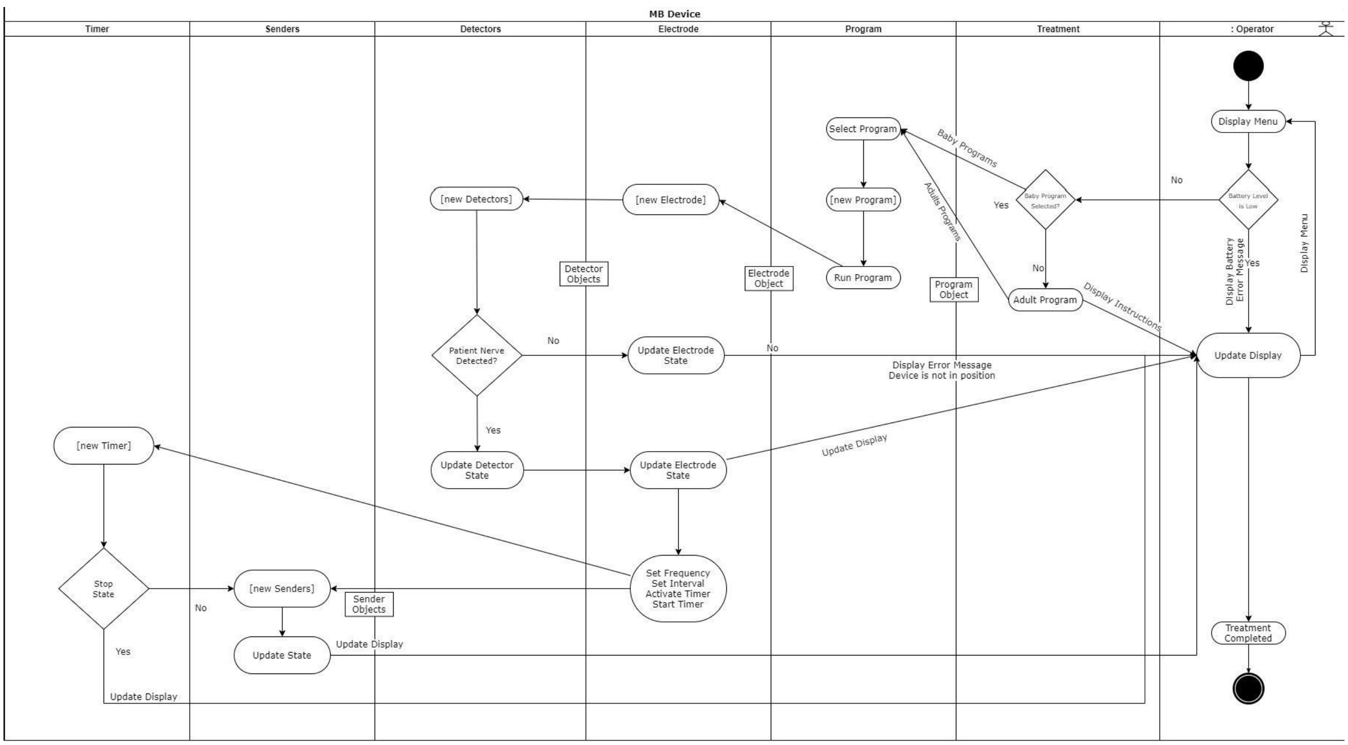
ready and is positioned on the correct contact point on the patient's body.

9- Once the Program object receive READY state from the Electrode' Detectors, it updates its object state and instructs the initialized Sender objects to start sending the treatment signals as modulated microcurrent as predefined in the program settings. Finally, the program object instructs the Interval object to count down the treatment duration.

10- Once the Program object receive READY state from the Electrode' Detectors, it updates its object state and instructs the initialized Sender objects to start sending the treatment signals as modulated microcurrent as predefined in the program settings. Finally, the program object instructs the Interval object to Activate the countdown of the treatment duration.

11- The Timer update the its state every second using the Ticking method, once the counter is less zero, it updates its state to STOP state. At this time, the Times states is sent back to the Interval objects which updates its state and sends back its state to the programs. Once the program object is acknowledged with the STOP state received, it instructs the Sends object to stop sending treatment signals and update the display screen, otherwise, the updates of the interval object is displayed on the screen showing the progress of the treatment.

12- Once the treatment program is completed acknowledge duration and STOP state is received from the timer, the Program object update the display screen which calls the device Update display method which finally triggers displaying the main menu waiting for next user activity.



MB Device
Timer
Senders
Detectors
Electrode
Program
Treatment
: Operator
Display Menu
Battery Level is Low
No
Yes
Display Battery Error Message
Display Menu
Baby Program Selected?
Select Program
Baby Programs
Adults Programs
[new Program]
Run Program
Program Object
Adult Program
Display Instructions
[new Electrode]
Electrode Object
[new Detectors]
Detector Objects
Patient Nerve Detected?
Update Electrode State
Display Error Message
Device is not in position
Update Display
Update Detector State
[new Timer]
Set Frequency
Set Interval
Activate Timer
Start Timer
Stop State
[new Senders]
Sender Objects
Update State
Treatment Completed

10. Usability Diagram

The diagram below depicts how the device will function in a way to deliver best user experience. This main usability object how to deliver best treatment functionalities and maintaining common device functions. In addition, the device has to be resilient to be integrated with wide variety external electrodes in the market ensuring best return on the investment to the customers.

10.1 Functionalities

10.1.2 Fast Response Time

The system architecture shall maintain best user experience with lowest response time. This can be achieved by initializing the required objects when it is needed. For instance, the time object will be initialized when then a Program is selected and the program object is initialized. This not only make best use of the device storage and memory, but also less processing needed and less power to consume.

10.1.2 Rich Treatment Programs

The main added value of using the system is the outcome of the Research and Development by discovering the best combination of microcurrent and frequency that can address medical conditions. The more the device is able to offer pain relief and treatments of variant of medical conditions, the more the device offers more usability.

10.1.3 Compatible with Existing Devices

Any system comes with software updates and versions. The system architecture should leverage existing hardware as much as possible to justify the investment for long-term use. The system design should be resilient to maximize the value of existing hardware as much as possible unless hardware is new and offers new hardware technology that does not functions with the system software.

10.1.4 Support Variety of Existing Electrode

Although the device has built-in Electrode, the system should be able to connect to external electrodes to cover wider body nerves. The device uses USB connectivity to recognize external electrode however the system software uses the same Electrode class to apply the activity and sequence design.

10.1.5 Support Variety of Detector

One crucial component is detecting right signal from the nerve. The system should be able to calibrate the detected volts and recognize the nerve signal associate with the treatment program.

10.1.6 Tuning Ability

The system offer predefines treatment programs as outcome from the RD. However, the system should be able to offer manual tuning for the microcurrent frequency and the treatment interval time to enable more flexible options.

10.1.7 Informative Instructions

The more the system is interacting with Operator, the more accurate treatment results. The predefined treatment program shall present to the operator descriptive and informative knowledge on the device's display to guiding the operator for best user experience.

10.2 Rich Function

10.2.1 Change System Setting

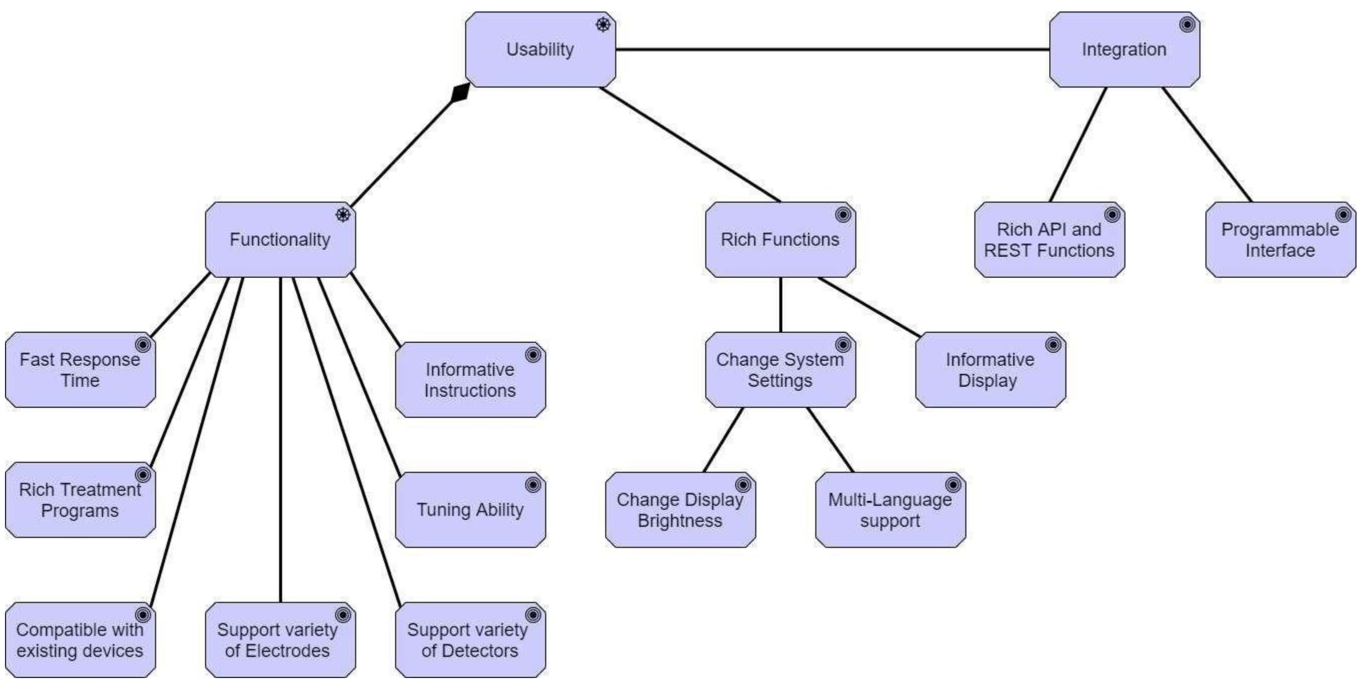
Big part of the usability is offering system utilities that enriches user experience. The system should be able to deliver ability of changing the device look and feel like brightness level and multi-language support.

10.2.2 Informative Display

System utilities should be able user-friendly and self-intuitive. The displayed information, the menu options should match the device's hardware capabilities. For instance, menu navigation should be use navigation keys and always there is an option to exit or going back to main menu.

10.3 Rich Integration

There are different types of user experience will not use the device, but seeking integrating their systems to the device. The system design should be flexible by offering integration package using existing calls to feedback these systems.


Usability
Integration
Functionality
Rich Functions
Rich API and REST Functions
Programmable Interface
Fast Response Time
Informative Instructions
Change System Settings
Informative Display
Rich Treatment Programs
Tuning Ability
Change Display Brightness
Multi-Language support
Compatible with existing devices
Support variety of Electrodes
Support variety of Detectors

11. Object-Oriented Analysis (OOA)

Base on the discovered classes in the analysis phase, The OOA design below depicts each class's functionality, properties and its execution method to maintain desired user experience and fast response time:

11.1 Aggregation

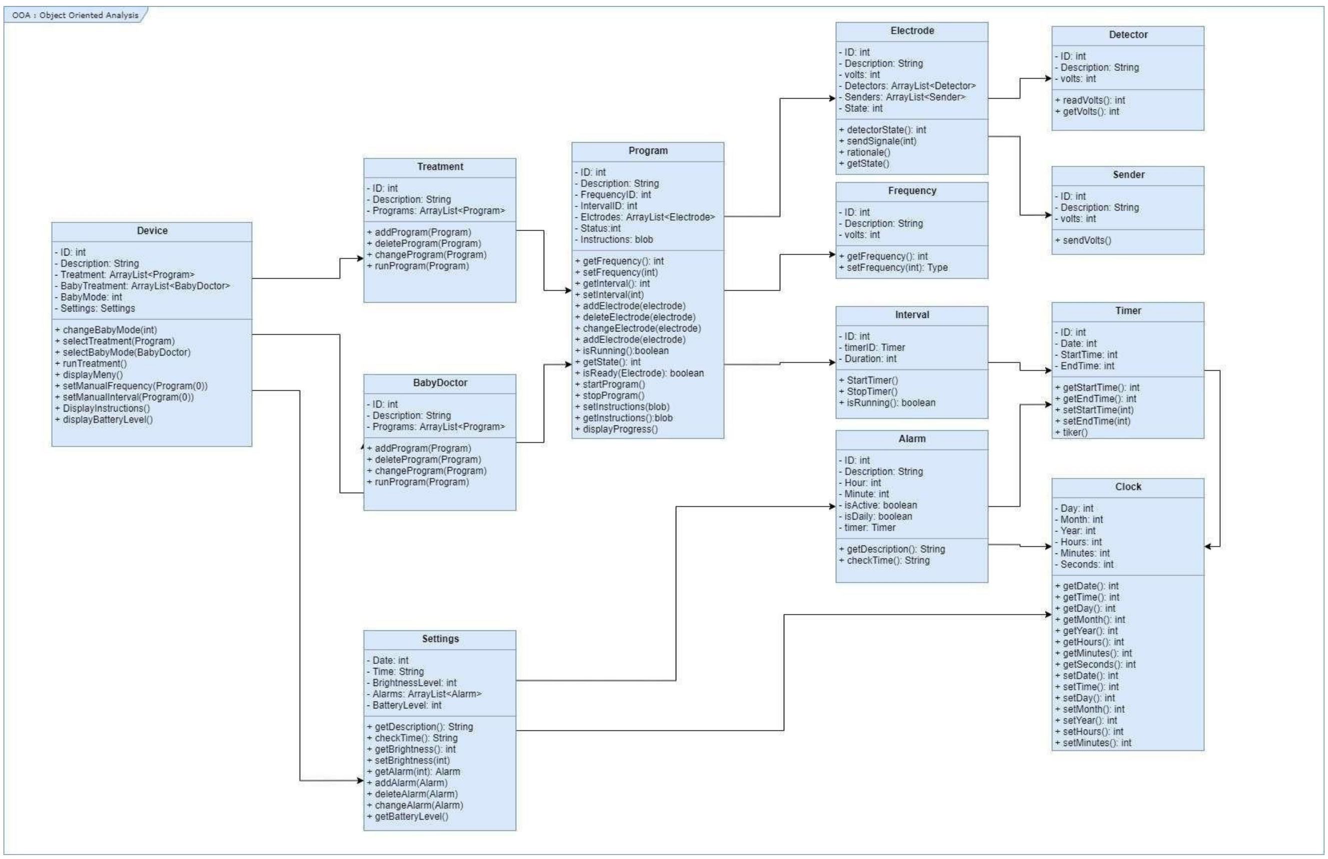
Class aggregation is used to ensure objects are initialize only when needed saving memory and processing utilization. For instance, the Program class has an aggregation of Electrode, Frequency, and Interval objects. This allowed the program having best use of the device's hardware only when the program in activated. Another example, the Interval class having Timer object where start and end times are captured from the system clock. If the device's detector is not ready, this timer object is disposed and initialized again by the Interval class. Additional example where Electrode class has Sender and Detector objects, to avoid conflicting signal from being running when the Elector is in a detection state making sure sender object is in idle mode.

11.2 Composition

The OOA addressed class dependencies especially in treatment programs. For instance, the device has four treatment modes for adults and baby age. This could lead having duplicated programs while age and frequency can be factorized. Classes are designed in a deposition ensuring dependency is not consuming processing.

11.3 Association

The OOA design below established association relationship through its Objects. Examples like Sender and Times objects where both are totally separate by are tightly associated based on the states, they send to the


OOA : Object Oriented Analysis
Device
- ID: int
- Description: String
- Treatment: ArrayList<Program>
- BabyTreatment: ArrayList<BabyDoctor>
- BabyMode: int
- Settings: Settings
+ changeBabyMode(int)
+ selectTreatment(Program)
+ selectBabyMode(BabyDoctor)
+ runTreatment()
+ displayMeny()
+ setManualFrequency(Program(0))
+ setManualInterval(Program(0))
+ DisplayInstructions()
+ displayBatteryLevel()
Treatment
- ID: int
- Description: String
- Programs: ArrayList<Program>
+ addProgram(Program)
+ deleteProgram(Program)
+ changeProgram(Program)
+ runProgram(Program)
BabyDoctor
- ID: int
- Description: String
- Programs: ArrayList<Program>
+ addProgram(Program)
+ deleteProgram(Program)
+ changeProgram(Program)
+ runProgram(Program)
Program
- ID: int
- Description: String
- FrequencyID: int
- IntervalID: int
- Elctrodes: ArrayList<Electrode>
- Status:int
- Instructions: blob
+ getFrequency(): int
+ setFrequency(int)
+ getInterval(): int
+ setInterval(int)
+ addElectrode(electrode)
+ deleteElectrode(electrode)
+ changeElectrode(electrode)
+ addElectrode(electrode)
+ isRunning():boolean
+ getState(): int
+ isReady(Electrode): boolean
+ startProgram()
+ stopProgram()
+ setInstructions(blob)
+ getInstructions():blob
+ displayProgress()
Electrode
- ID: int
- Description: String
- volts: int
- Detectors: ArrayList<Detector>
- Senders: ArrayList<Sender>
- State: int
+ detectorState(): int
+ sendSignale(int)
+ rationale()
+ getState()
Frequency
- ID: int
- Description: String
- volts: int
+ getFrequency(): int
+ setFrequency(int): Type
Interval
- ID: int
- timerID: Timer
- Duration: int
+ StartTimer()
+ StopTimer()
+ isRunning(): boolean
Alarm
- ID: int
- Description: String
- Hour: int
- Minute: int
- isActive: boolean
- isDaily: boolean
- timer: Timer
+ getDescription(): String
+ checkTime(): String
Detector
- ID: int
- Description: String
- volts: int
+ readVolts(): int
+ getVolts(): int
Sender
- ID: int
- Description: String
- volts: int
+ sendVolts()
Timer
- ID: int
- Date: int
- StartTime: int
- EndTime: int
+ getStartTime(): int
+ getEndTime(): int
+ setStartTime(int)
+ setEndTime(int)
+ tiker()
Clock
- Day: int
- Month: int
- Year: int
- Hours: int
- Minutes: int
- Seconds: int
+ getDate(): int
+ getTime(): int
+ getDay(): int
+ getMonth(): int
+ getYear(): int
+ getHours(): int
+ getMinutes(): int
+ getSeconds(): int
+ setDate(): int
+ setTime(): int
+ setDay(): int
+ setMonth(): int
+ setYear(): int
+ setHours(): int
+ setMinutes(): int
Settings
- Date: int
- Time: String
- BrightnessLevel: int
- Alarms: ArrayList<Alarm>
- BatteryLevel: int
+ getDescription(): String
+ checkTime(): String
+ getBrightness(): int
+ setBrightness(int)
+ getAlarm(int): Alarm
+ addAlarm(Alarm)
+ deleteAlarm(Alarm)
+ changeAlarm(Alarm)
+ getBatteryLevel()

12. Traceability matrix

Base on the discovered classes in the OOA, the design below show requirement will be traces as follow:

REQUIREMENTS TRACEABILITY MATRIX							
Project Name: Microcurrent Biofeedback Device							
Business Requirements Document BRD		Functional Specification Document FSD			Test Case Document		
Business Requirement ID#	**Business Requirement I Business Use case**	**Functional Requirement ID#**	**Functional Requirement I Use Case**	**Priority**	**Test Case ID#**	**Execution Status** Pass/Fail	**Defect#**
BR_1	Device Module	FR_1	Display Menu	High	TC#001 TC#002		
		FR_2	Load system settings	High	TC#003 TC#004		
		FR_3	Update Display	High	TC#005 TC#006		
		FR_4	Load Treatment Module	High			
BR_2	Treatment Module	FR_5	Populate Programs	High	TC#007 TC#008		
		FR_6	Miniplate Programs	High	TC#009		
		FR_7	Run Program	Medium	TC#010 TC#011		
BR_3	Program Module	FR_8	Miniplate Electrode	High	TC#0012 TC#0013		
		FR_9	Miniplate Frequency	High	TC#0014 TC#0015		

		FR_10	Miniplate Interval	High	TC#0016 TC#0017		
		FR_11	Monitor and update status	High	TC#0018 TC#0019		
BR_4	Electrode Module	FR_12	Miniplate Detectors	High	TC#0020 TC#0021		
		FR_13	Miniplate Senders	High	TC#0022		
		FR_14	Monitor and update status	Medium	TC#0023 TC#0024		
BR_5	Detector Module	FR_15	Miniplate Detector	High	TC#0025 TC#0026		
		FR_16	Monitor and update status	High	TC#0027 TC#0028		
BR_5	Sender Module	FR_17	Miniplate Sender	High	TC#0029 TC#0030		
		FR_18	Monitor and update status	High	TC#0031 TC#0032		
BR_6	Interval Module	FR_19	Miniplate Interval	High	TC#0033 TC#0034		
		FR_20	Miniplate Timer	High	TC#0035		
		FR_21	Monitor and update status	Medium	TC#0036 TC#0037		