mainwindow

+ bool flag

- UI::MainWindow *ui
- Device* device

- int selectedSession
- int selectedGroup
- int connectionIntensity
- int timesIntensityAdjusted
- bool endSessionEarlyFlag

- QElapsedTimer blinkTimer
- QElapsedTimer powerPressedTimer
- QElapsedTimer selectTimer
- QElapsedTimer pauseTimer
- QElapsedTimer therapyTimer
- QElapsedTimer timeoutTimer
- QElapsedTimer testingTimer

- therapy(int groupNum, int sessionNum, int recordingFlag, int overrideIntensity)
- connectionTest()
- addRecording(const string& name, int group, int batteryPercent, int initialIntensity, int intensity)
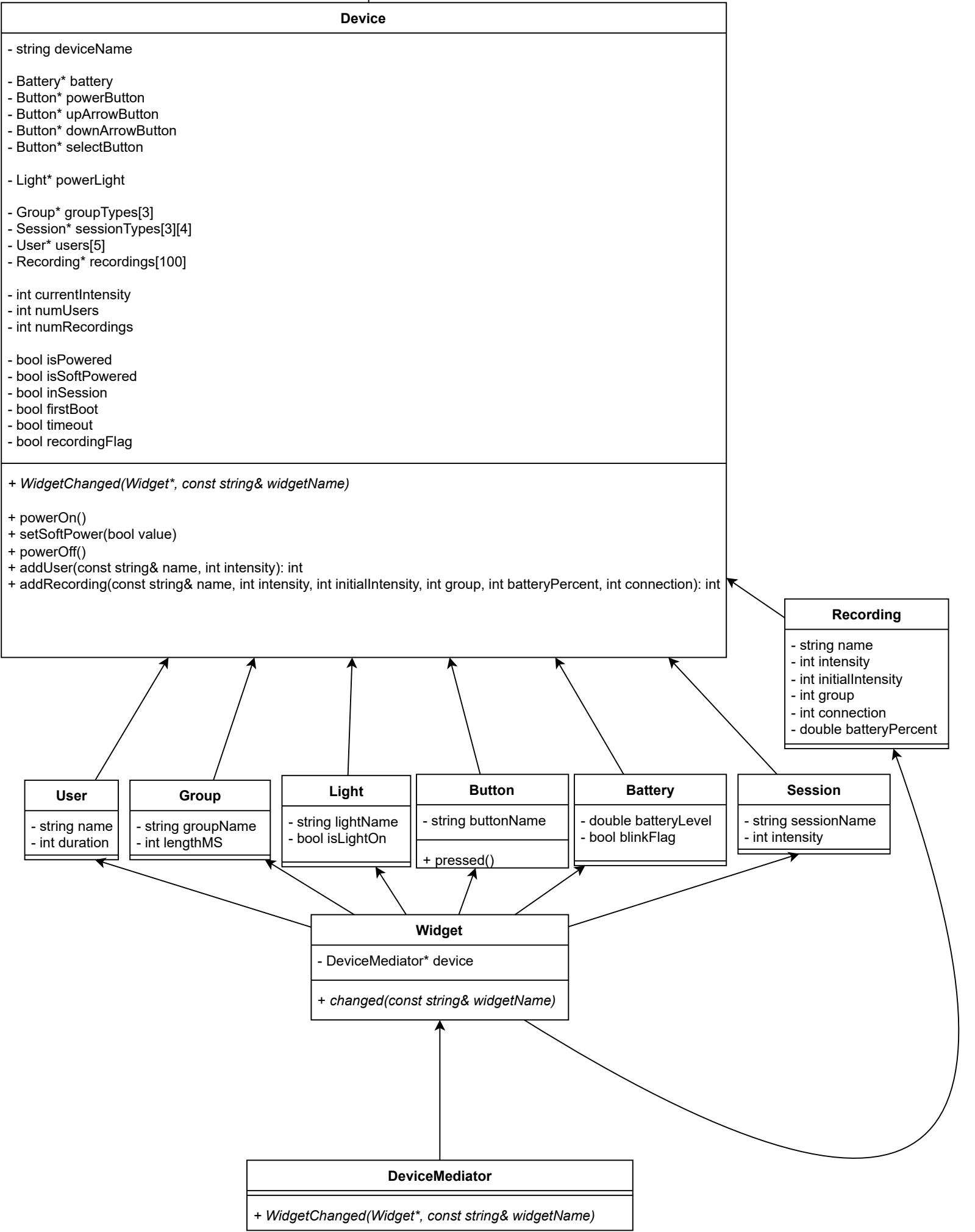- replayRecording(Recording* recording)

- changeTextColor(QTextBrowser *text, QColor color)
- changeBackgroundColor(QPushButton *button, const QString& color, const QString& image, const QString& radius)
- bootConnectionTest()
- blinkSession(int sessionNum)
- cycleGroupButton()
- turnOffUI()
- softOff()
- setTherpayLock(bool status)
- updateIntensityLog()
- displayIntensityOnGraph()
- powerLightOnOff(bool status)
- endSessionEarly()

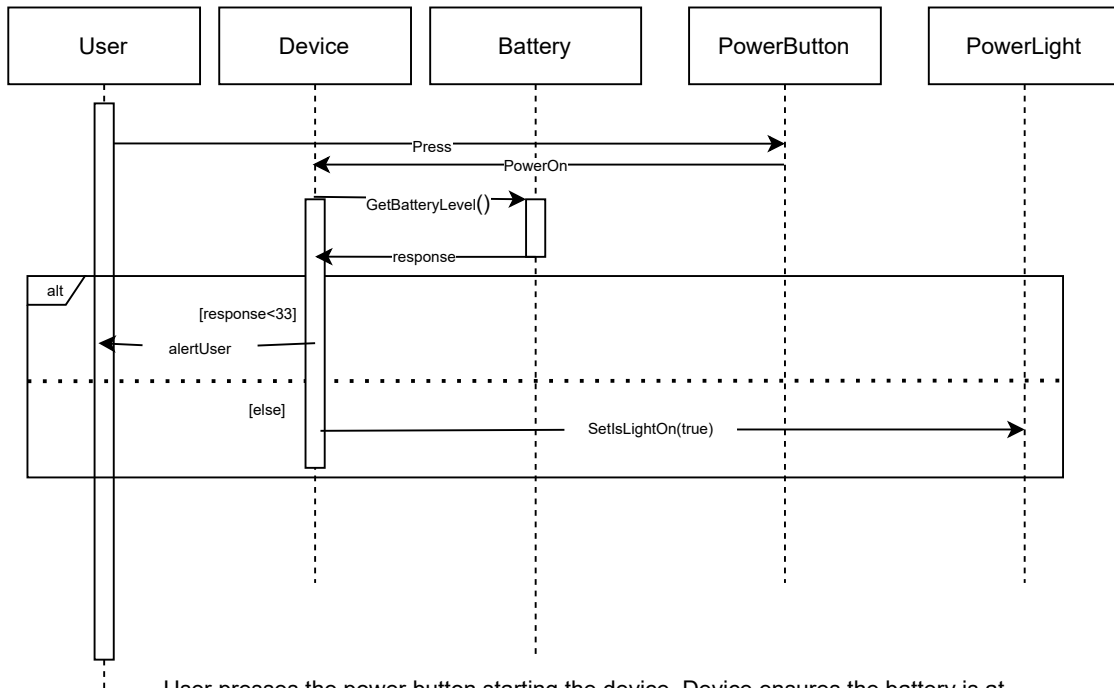- ui_initializeBattery()
- ui_turnOffBattery()

- blinkBattery()
- drainBattery(int intensity)
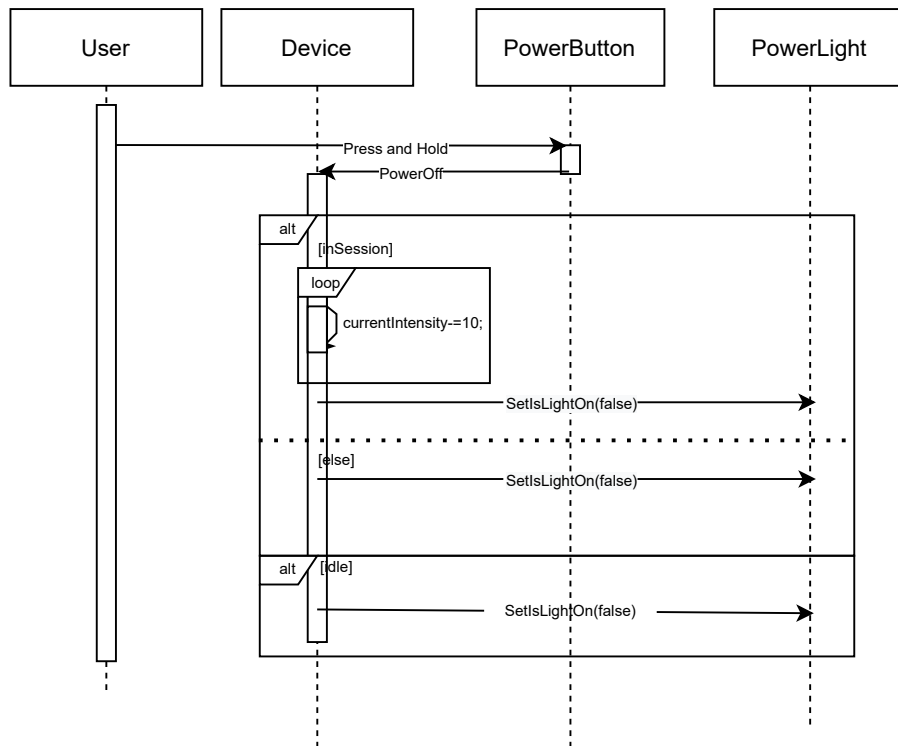- checkBattery(): bool
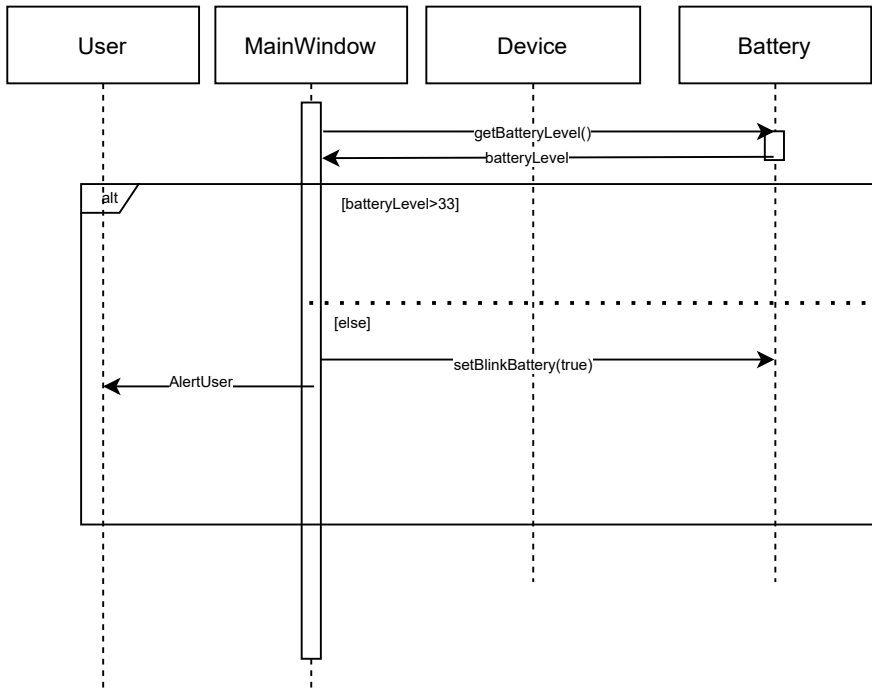- batteryWarning()

- sleepy(int sleepTime)

Device

- string deviceName

- Battery* battery
- Button* powerButton
- Button* upArrowButton
- Button* downArrowButton
- Button* selectButton

- Light* powerLight

- Group* groupTypes[3]
- Session* sessionTypes[3][4]
- User* users[5]
- Recording* recordings[100]

- int currentIntensity
- int numUsers
- int numRecordings

- bool isPowered
- bool isSoftPowered
- bool inSession
- bool firstBoot
- bool timeout
- bool recordingFlag

+ *WidgetChanged(Widget*, const string& widgetName)*

+ powerOn()
+ setSoftPower(bool value)
+ powerOff()
+ addUser(const string& name, int intensity): int
+ addRecording(const string& name, int intensity, int initialIntensity, int group, int batteryPercent, int connection): int

---

Recording

- string name
- int intensity
- int initialIntensity
- int group
- int connection
- double batteryPercent

User

- string name
- int duration

Group

- string groupName
- int lengthMS

Light

- string lightName
- bool isLightOn

Button

- string buttonName

+ pressed()

Battery

- double batteryLevel
- bool blinkFlag

Session

- string sessionName
- int intensity

Widget

- DeviceMediator* device

+ *changed(const string& widgetName)*

DeviceMediator

+ *WidgetChanged(Widget*, const string& widgetName)*

Power On

User	Device	Battery	PowerButton	PowerLight

User → PowerButton: Press
PowerButton → Device: PowerOn
Device → Battery: GetBatteryLevel()
Battery → Device: response

alt
[response<33]
Device → User: alertUser

[else]
Device → PowerLight: SetIsLightOn(true)

User presses the power button starting the device. Device ensures the battery is at
acceptable levels, if it is the device starts and lights are turned on.
If it isn't the user is prompted to change batteries. Refers to the Power On use case (UC1).

Power Off

User	Device	PowerButton	PowerLight

User → PowerButton: Press and Hold
PowerButton → Device: PowerOff

alt
[inSession]
loop
Device: currentIntensity-=10;
Device → PowerLight: SetIsLightOn(false)

[else]
Device → PowerLight: SetIsLightOn(false)

alt
[idle]
Device → PowerLight: SetIsLightOn(false)

The user presses and holds the power button, signaling the device to shutoff.
If it is running a session the intensity is slowly lowered before shutting off.
Alternatively, if the device has been idle for too long after booting the device shuts off. Refers to the Power Off use case (UC2).

Low Battery

User	MainWindow	Device	Battery

getBatteryLevel()

batteryLevel

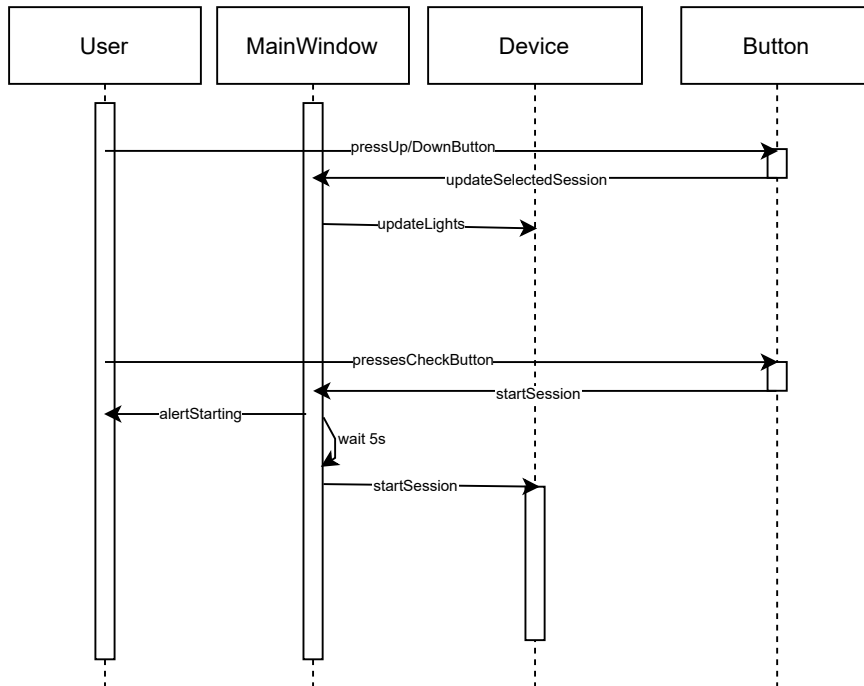**alt** [batteryLevel>33]

[else]

setBlinkBattery(true)

AlertUser

The main window periodically checks the battery level.
If battery levels are sufficient functioning continues, else the battery light blinks and user is prompted to change batteries.
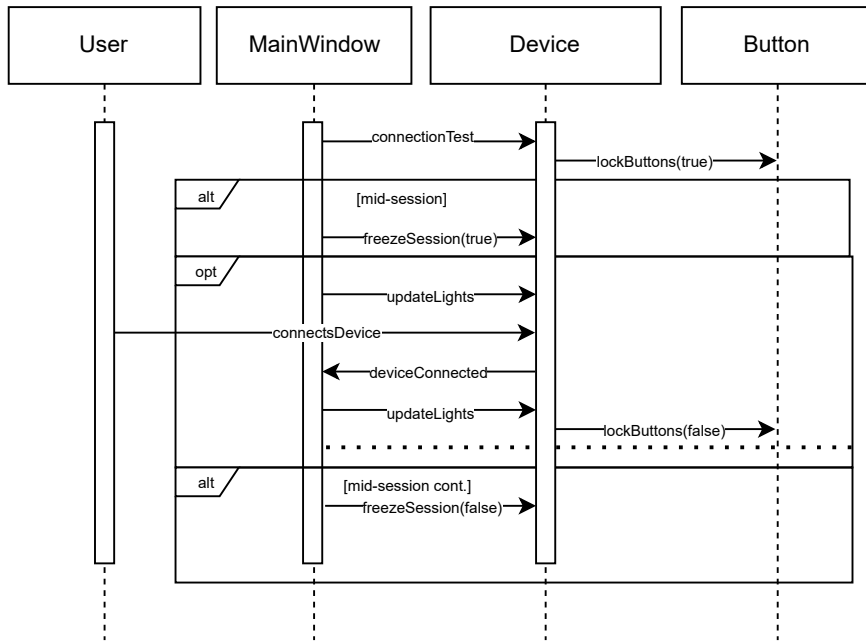Refers to the Low Battery use case (UC3)
.

Selecting a Session

User	MainWindow	Device	Button

pressUp/DownButton

updateSelectedSession

updateLights

pressesCheckButton

startSession

alertStarting

wait 5s

startSession

User wants to select a session, presses the up/down buttons to change selection.
Main window is updated as to which selection is active, lighting up on the device.
After user confirms session, it alerts the user then starts after 5 seconds. Refers to the Selecting a Session use case(UC4)
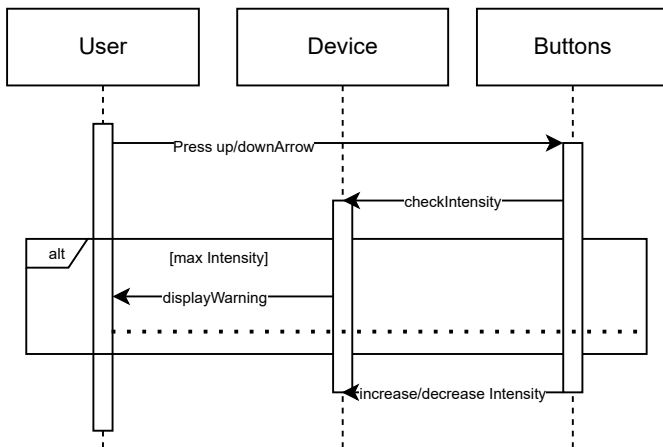.

Connection Test

User	MainWindow	Device	Button

connectionTest

lockButtons(true)

alt [mid-session]

freezeSession(true)

opt

updateLights

connectsDevice

deviceConnected

updateLights

lockButtons(false)

alt [mid-session cont.]

freezeSession(false)

The connection test begins upon startup or if the device is disconnected during a session. If the device is disconnected, buttons will be disabled until a connection is re-established. The mid-session alternative shows the session being frozen until a connection is re-established. If not mid-session, the device blinks the intensity lights to indicate it is looking for a connection. It continues this behaviour until a connection is established.
This refers to the connection test use case (UC5)

Intensity

User	Device	Buttons

Press up/downArrow

checkIntensity

alt [max Intensity]
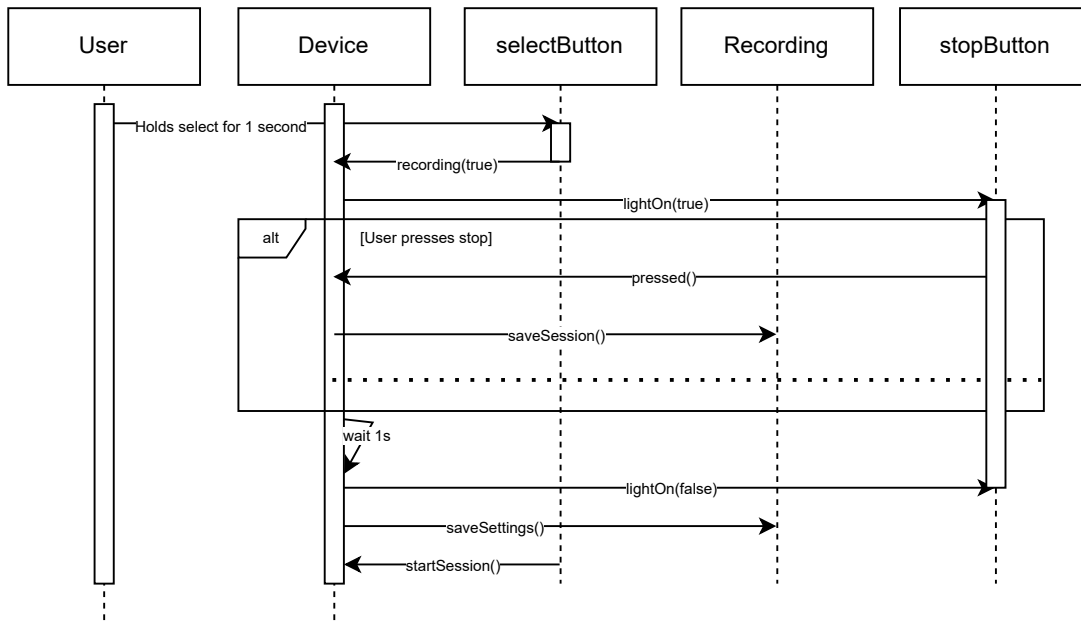
displayWarning

increase/decrease Intensity

When the user presses the up/down arrow button during or while configuring a session, the intensity is increased or decreased.
The device first checks that the desired intensity is within the range or possible intensities and if not, a warning is displayed.
This refers to the Intensity use case (UC6)

Record

If the user holds the select button for a second they prime the device to remember the current settings.
When this happens, the user can choose to press 'Stop' which saves the settings but does not execute the session.
Otherwise the session's settings are saved and the session is executed.
This refers to the Record use case (UC7)

Replay
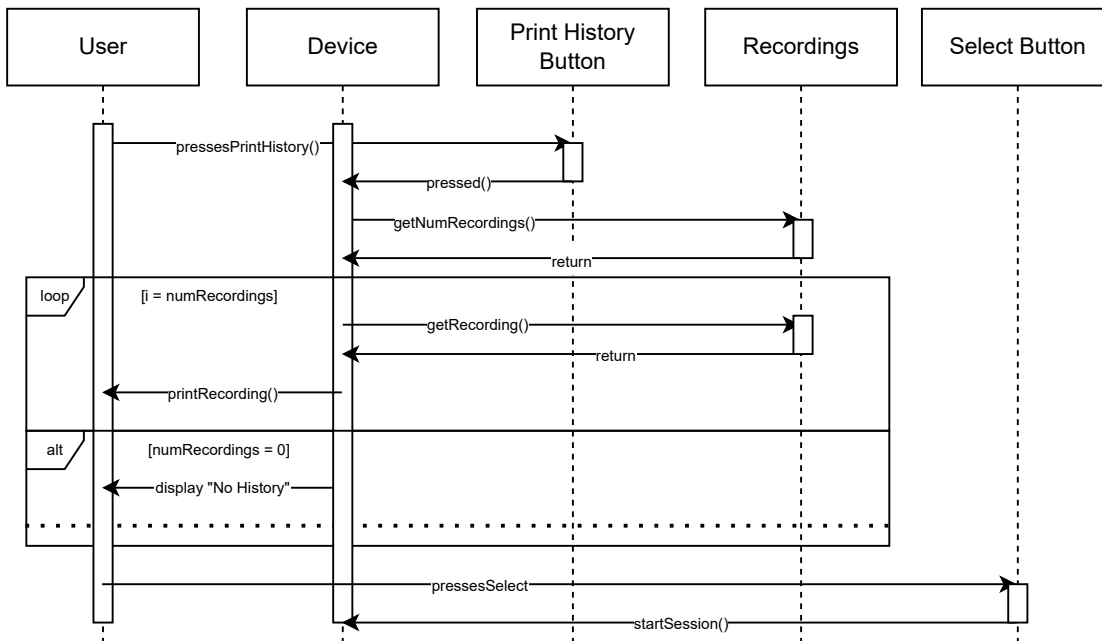
If the user presses the Print History Button, the device goes through the user's recording sessions and displays their settings.
If the user has no recorded sessions the device displays "No History".
Otherwise the user may select a recorded session, then press select to begin a session with the recorded settings.
This refers to the Replay use case (UC8)