**Test Plans [DRAFT]**

Title: New User Design Flow

Case Number: 001

Purpose: Run through app as a new user

Steps:

* Open app
* Registration
* Login
* Add User

Expected Results: User should be able to register Lumi ID and set up password. Login with new credentials. Create a new User name attached to Lumi ID.

Title: Registered User

Case Number: 002

Purpose: To show registered user saved settings

Steps:

* Open App
* Skips Login screen
* Choose User from saved User List
* Open Message Board with saved Message Log
* Check Settings to see saved user configuration

Expected Results: User should open app and directs to Main Menu. All settings configured prior to

last use should still be saved.

Title: New User Error Checking

Case Number: 003

Purpose: Run through app as a new user triggering error conditions

Steps:

* Open app
* Try to Login with empty fields
* Try to Login with wrong credentials
* Register with empty fields
* Register with wrong Lumi ID
* Register with mismatching passwords
* At main menu, navigate to new screen without choosing user

Expected Results: Attempting to login with empty fields or wrong credentials will Toast associated

error message. Registering with empty fields, wrong Lumi ID or mismatching passwords.

Title: Using the Message Board

Case Number: 004

Purpose: Testing the Message Board functionality

Steps:

* Enter message text in Message Board as User 1
* Enter message text in Message Board as User 2
* Trigger Low Light Level Event
* Trigger Crying Event
* Restart App

Expected Results: Message Log should show message text from which user with time stamp chronologically. Also show Baby Menu events displayed with time stamps.

All are saved to database dynamically which is demonstrated from leaving

activity, restarting app, or loading app by different user.

Title: Using Baby Menu

Case Number: 005

Purpose: Testing Baby Menu

Steps:

* Enable Auto Mode
* Enable Crying Mode
* Choose Lullaby
* Restart App
* Turn off Lights
* Yell in microphone

Expected Results: Lumi should save user configuration in database. Upon Low light levels, trigger event will be appended to message log, and phone will vibrate. Upon high decibel levels, lullaby will play on phone.

Title: Online/Offline Testing

Case Number: 006

Purpose: Using the app Online and Offline

Steps:

* While Offline, Open App
* Change 1 setting in Settings
* Enter message text in Message Board
* Close App
* While Online, Open App
* Enter message text in Message Board

Expected Results: While Offline, user will be notified nothing will be updated until Online. Configuration and Message Log will be saved to local database. Upon opening app Online, local database dynamically appended to user configuration and logs.

Title: Color Palette

Case Number: 007

Purpose: Testing Color Palette Screen

Steps:

* Press Color Button
* On Click, change demo image color to selected color
* Click on custom color to set own color

Expected Results: User is able to change color or set a color they choose to demonstrate color selection change. Also, able to save custom colors for personalization.

Title: Timer Menu

Case Number: 008

Purpose: Testing timer menu

Steps:

* Open Timer Menu
* Set Timer
* Set Alarm
* Save Preset
* Wait for Timer
* Wait for Alarm

Expected Results: Set timer and alarms as one time or custom. Also, should work in tandem with Notifications.

Title: Media Player

Case Number: 009

Purpose: Test Media Player Functionality

Steps:

* Open Baby Menu
* Open Media Player Dialog Box
* Play preset songs
* Add songs from phone storage

Expected Results: User should be able to play songs, add custom songs to trigger events.

Title: All Error Condition Triggering

Case Number: 010

Purpose: Checking all error conditions

Steps:

* Open App
* Invoke Registration Screen errors
* Invoke Login Screen errors
* Invoke Main Menu errors
* Invoke Baby Menu Screen errors
* Invoke Message Board Screen errors
* Invoke Settings Screen errors
* Invoke Notification Screen errors
* Invoke Timer Screen errors
* Invoke Color Palette errors
* Invoke Fetch/Send database errors

Expected Results: Should demonstrate each hard-coded error by doing conditional tests to invoke error messages.