

Case Study

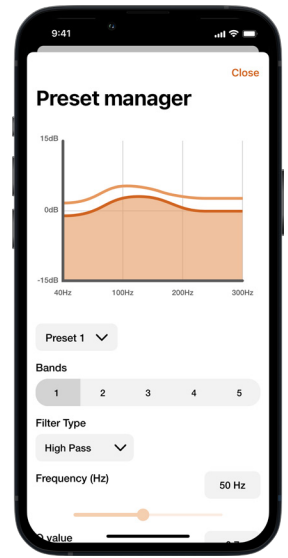
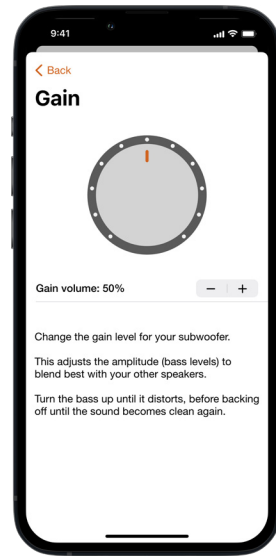
Elytone - MVP iOS App



The purpose of the project was to create a MVP iOS and Android application which would interact with a smart Subwoofer over BLE. The application would be updated over time to support new products as they were added to Elytone's subwoofer range. It allowed the user to change device settings such as volume, dynamic bass and phase. As well as more advanced features such as room correction and firmware updates.

At a glance

- Native iOS and Android application development.
- Design and development of a BLE companion app interacting with a subwoofer.
- Embedded firmware development using Cypress.
- Utilized a reusable communication protocol (Device Interaction Protocol) to expand device features efficiently.
- Introduced a state-of-the-art room correction system using the phone's microphone.
- Development of an over-the-air firmware update process.



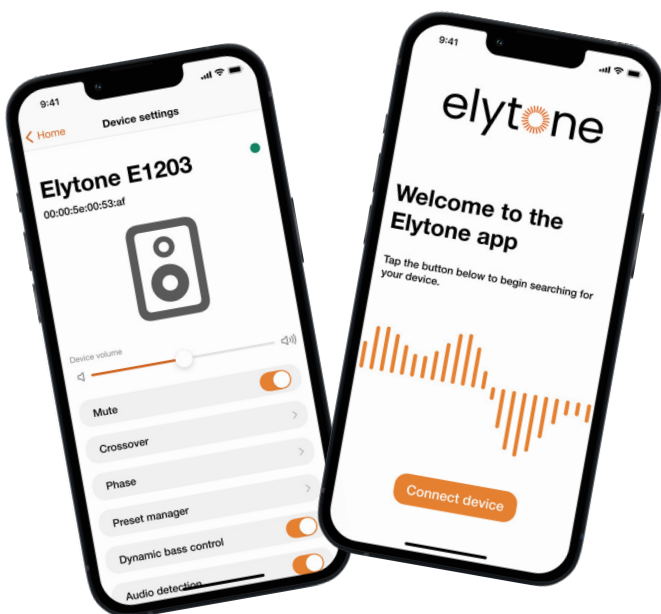
The Challenge

Elytone, design and manufacture electronic components and solutions for audio products. They were looking to introduce a new line of subwoofer components to their product range. With the ability to adjust various subwoofer settings over BLE from an iOS or Android companion app.

These subwoofer components and mobile app would serve as the foundation for other audio manufacturers to build their own products. This meant that the communication protocol needed to be extensible so that new commands could be added over time to support future subwoofer models with new capabilities as well as custom commands added for specific hardware manufacturers.

The mobile app was required to support advanced features such as an over-the-air firmware update feature as well as a cutting-edge room correction feature.

The client chose Coderus due to our large amount of experience designing and developing bluetooth applications in the IoT spaces. We were tasked with designing the full solution including producing the UI designs, developing the mobile application and embedded firmware.



To find out more visit [Coderus.com/case-studies](https://www.coderus.com/case-studies)



The Methodology

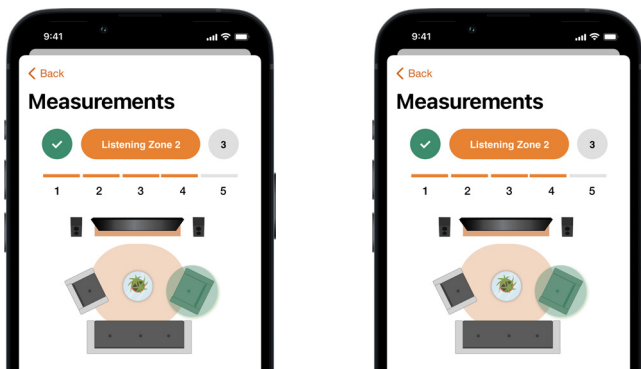
The project followed an agile SCRUM methodology with 2 week sprints to ensure a continuous feedback loop between us and the client. The client was involved planning and review sessions throughout the project. At the end of each sprint a new release was made available with the work completed so far.

The UX designers, developers and QA all worked on the same sprint. Helping to ensure that designs were completed and ready for development and QA were aware of what features had been planned into the sprint and when they would be completed.

The QA team verified each new feature as they were developed. This close collaboration between the developers and QA meant we were quickly able to identify and resolve bugs or other issues as early as possible, before code changes were merged into the main codebase. This approach of testing regularly and often meant the application was more robust and had less overall unexpected behaviour when delivering builds to the client.

Because we designed the user interface in-house, this allowed us to work closely with the designers to ensure the app flow worked well with the underlying technical steps required by the hardware, as well as following platform conventions.

Delivering a solution quickly was a requirement from the client, so we utilised the latest Swift and iOS technologies such as SwiftUI to rapidly prototype the user interface, and various library features to get the app off the ground quickly. By targeting a later version of iOS, we utilised newer frameworks such as Swift Charts to create graphs quickly.



The Android app made use of Jetpack Compose the new UI framework from Google to quickly build the screens. Coderus reused a BLE library developed in house to allow BLE interaction with the subwoofer to be quickly developed.

Elytone asked us to work with an audio algorithm supplier who developed the model which worked with the recordings of the subwoofer captured by the mobile app to identify the optimisations which should be applied to the subwoofer. We held a number of workshop sessions to allow us to quickly iterate on the feature to ensure correct behaviour and calibration.



The Results

We delivered an initial MVP of the app to the client as part of a continuous delivery process. We delivered in bi-weekly sprints. The app features a pairing flow, a list of the user's already saved devices, a general settings screen, device volume control, dynamic bass boost, EQ presets, phase control, room correction and firmware updates.

All our BLE logic and RoomEQ logic is in separate libraries so can be developed separately or be reused in other applications.

We saved time by using the latest UI technologies, allowing us to create the app screens and graphs faster.

The Tech

- |  iOS |  Android |
|---|---|
| <input type="checkbox"/> Swift & SwiftUI | <input type="checkbox"/> Kotlin |
| <input type="checkbox"/> Swift Package Manager | <input type="checkbox"/> Jetpack Compose |
| <input type="checkbox"/> Swift Concurrency | <input type="checkbox"/> MVVM |
| <input type="checkbox"/> Dependency Injection / Swinject | <input type="checkbox"/> Kotlin Coroutines |
| <input type="checkbox"/> Core Bluetooth | <input type="checkbox"/> Dependency Injection using Hilt |
| <input type="checkbox"/> Swift Charts | <input type="checkbox"/> Architecture Components |
| | <input type="checkbox"/> BLE |

To find out more visit Coderus.com/case-studies

