Submission Demos track for NordiCHI 2022

Hearing Aid adjustment interface design and development

A case project of Computer Science (AP) selective course “UI design”

Xiaolei, XBI, Bi

Adjunkt, Erhvervsakademi Midtvest, xbi@eamv.dk

Kiryl, KIRY, Zhukau, Student, Erhvervsakademi Midtvest, [kiry0001@eamv.dk](mailto:%20kiry0001@eamv.dk)

Yusuf, YUSU, Aslantas, Student, Erhvervsakademi Midtvest, [yusu0215@eamv.dk](mailto:yusu0215@eamv.dk)

Christian, CHRI, Christensen, Student, Erhvervsakademi Midtvest, [chri11g6@eamv.dk](mailto:chri11g6@eamv.dk)

The students got the task to redesign an interface for a hearing aid brand. After user research, the students found design solution to an existing problem where the hearing aid users could get a notification when the battery is low. The user interface was designed and then prototype was made and ready for test.

UI design •Prototype • Web APP

**Additional Keywords and Phrases:** BEM methodology, Sass - Syntactically Awesome Stylesheet, SPA (Single Page Application)

1. Introduction

The students are reading Datamatiker AK (Erhvervsakademi Midtvest Herning) (AP Graduate in Computer Science) in their 4th semester. They had a selective course in this semester ‘UI Design and development’ where they were challenged to work on a user interface redesign and prototype building.

They primary used design thinking to scope this project. Personas and scenarios were developed, and paper prototype was made. Then the concept was developed and built.

* 1. Project Kick off

The project was kicked off by a remote meeting from a hearing aid company in Copenhagen with a short introduction about the hearing aid and fitting session. Shortly after the presentation, the students made field study to observe hearing aid users.

* 1. Define

By the end of the first week, personas and custom user journey mapping was created. Using these methods, students figured out the pain and gains of hearing aid patient.

* 1. Ideate

The students started with brainstorming and then drew the paper prototypes. They designed from Low-Fi prototype to Hi-Fi prototype and discovered accessibility issues to improve for the design use.

1. Prototype

When the design proposal was approved, the students continued with coding.

Prototype Link

[Case 2 – Figma](https://www.figma.com/file/fdy13igEQnyZ6LloX0Todn/Case-2?node-id=0%3A1)

* 1. Coding versions

The design was developed in two different versions, using different technologies. The first version is developed with pure HTML, CSS and a bit of JavaScript, and SASS is used to generate CSS, to make the CSS structure cleaner, and easier to read and further develop. The second version was developed using the React library. Both versions were visually identical, however, the React version had slightly more features.

React version can be found in the following link:

<https://go-ly.netlify.app/>

Links to source code:

HTML and CSS - <https://github.com/kotikkir9/hearing-aid-app>

React - <https://github.com/kotikkir9/hearing-aid-react-app>

For naming CSS classes, BEM methodology was used, which stands for Block, Element, Modifier. This means that HTML elements got more meaningful class names that were easy to find, understand and maintain. An example might look like this, header\_\_btn — green, Block as header, Element as a button, and Modifier as the green version of the button to use.

To generate CSS code, Sass (syntactically awesome style sheets) was used, which was preprocessor scripting language, and it was written with SCSS syntax- similar to the regular CSS syntax. The advantage of using Sass was that it had features like nesting, variables (found in CSS, however, Sass variables were much more flexible), mixins (flexible and reusable CSS declarations), modules, functions and more. These features made the code for styling much more structured, easy to read, easy to maintain and further develop.

The second version of the application was developed using the React library. The reason for this was the hearing aid application designed as a mobile app. With React it would be possible to develop a SPA (Single Page Application)- a web application that can run in any browser. The result was that each page of the application would be not a static HTML page. Each page would be generated using JavaScript on the client side (in browser), and the client did not have to ask the server for a new HTML page every time a page shift occurs. React uses JSX (syntax extension for JavaScript) to render UI. In addition, React Native could also have been used, as it, like React, uses JSX to render a user interface, however, React Native compiles the code for a real mobile application that can be used on both Android and iOS devices.