1 State of the art

In recent years, our healthcare service has been heavily tested; chances are it will be tested many times more. Numerous people have suffered, yet as a society, we have come out of this crisis stronger than ever. Therefore the opportunity to put my bit into the work and help seemed like a perfect goal of my bachelor's thesis.

I participated in an international project focused on stroke care quality registration, also known as RES-Q [1]. The desired effect of the project was to help registered providers improve their stroke care system. The project's main result was a web application. It was collecting stroke-related healthcare data, processing and evaluating them, and providing analysis and feedback. Healthcare providers participating in this project received awards based on the results of the project's evaluation. They could display their received awards as proof of their professionality and quality.

When I joined the project, the old version of the application had been running for a while, and it couldn't be easily maintained or updated. The user interface needed to be modernized and clarified. Another problem was that it was used only to collect stroke-related data. The project's goal was to develop a better version of the application and create a more stylish and functional user interface design using knowledge gained from the old application in the process. The new generation of the application was supposed to be ready for expansion. It had to be done in a more general way so that it could be used in other medical domains, not just stroke-related one.

My particular piece of work was the evaluation part. Many user roles were using the old application, and the new version should have included all of them. There were doctors, who collected the data, local coordinators representing healthcare providers, and group coordinators who could approve or edit the evaluation results. On top of that, they should have been provided with some extra evaluation functionalities in the new version of the application. Doctors and local coordinators should have been able only to see the results and ask for a recalculation. Group coordinators should have been able to see all pending proposals for awards, approve or edit them and make a note.

In addition, they should have been able to see all already evaluated providers.

Because of their different access to the evaluation functionalities, I had to make a clear separation between them. However, some components of the user interface were used in all the views. Therefore, with the abilities of some of the best design tools and front-end frameworks [2] available, such as React and Next.js [3], I had to create general components that could be used across all of the views without the possibility of unauthorized use.

Introduction

The state of healthcare services in 2020 and the following years was more important than ever. One of the pieces that could make a difference in this area was the worldwide stroke care quality improvement platform, also known as the Registry of Stroke Care Quality (RES-Q) [1]. Its goal was to collect stroke-related data, evaluate it, and provide comprehensive analyses and feedback. By the end of the year 2022, it was used at over 2000 hospitals by more than 550 000 patients in 91 countries. However, the state of the application processing the data was considerably old and needed some improvements.

This bachelor thesis aimed to create the evaluation part of the user interface of the new generation of the application. It had to provide a display of the results of the evaluation, allowing certain users to change the results manually and providing coordinators responsible for approving the results a way of communicating with each other in the form of notes. All of that had to be done in a general way so that it could have been be used in other medical domains, not just stroke-related one. On top of that, every component of the evaluation user interface had to be dynamic so that users were instantly able to see the results of their actions.

The first task was to create wireframes in Figma. I was constantly in touch with professionals from both the programming and user experience areas of expertise. All of them were providing feedback on the prototypes and wireframes I created. After multiple reworks and discussions with them, a clear goal of what my part of the application was supposed to be doing was set. I was able to start programming.

At first, I had to create fake data that I used during my work, because the backend team that was working on providing the actual data still needed to implement their solution. I used the React library and NextJS framework to make the pages and components, such as tables, forms, and text areas. After the backend team provided their solution, I used use real data in my application. The last step was to prepare the evaluation part for internationalization and properly connect it to the whole RES-Q web application.

The result of the bachelor thesis was a complete evaluation part of the user interface of the RES-Q web application. It provided tables

filled with the results of healthcare providers connected to the RES-Q platform and allowed coordinators to approve the evaluation results and communicate with each other. All of the components used in the pages related to evaluation were ready to be used in other medical domains and also ready to be used in other languages.

2 About the project

Registry of stroke care quality

(https://resq.netlify.app/about/history) Registry of stroke care quality, also known as RES-Q, is a project developed by ESO EAST (European Stroke Organisation: Enhancing and Accelerating Stroke Treatment) primarily targeted at Central and Eastern Europe. On July 7th, the RES-Q project crossed an important milestone of 500 000 stroke patients registered in the RES-Q database from over 90 countries worldwide.

(https://resq.netlify.app/about/history) (https://resq.netlify.app/about/vision-and-mission)

The primary goal of RES-Q is to improve the quality of stroke care in central and eastern Europe by providing data that can be translated into healthcare policies and guidelines on both a national and European level. The desired effect of the project is not to find new treatments but to improve the implementation of already existing ones.

(SYSTEM ARCHITECTURE) RES-Q collects as much healthcare-related data as possible calculates healthcare quality metrics, and provides comprehensive analyses, feedback, and evaluation to the user in an easily understandable way.

3 System architecture

User Roles

Data Entry The Data Entry role collected medical records for a specific provider. Its access to the Evaluation part of the user interface was limited and contained access to Received Awards. In real life, it might have been a doctor.

Local Coordinator The Local Coordinator's role was to represent a healthcare provider. Its key responsibility was communicating with other coordinators and administrators, assigning Data Entry roles to new users, and updating providers' profiles. It had the same access to the Evaluation as the Data Entry role.

Group Coordinator The Group Coordinator communicated with all other coordinators and administrators and approved new providers into the group. Its access to the Evaluation part of the user interface included access to the Pending Proposals page and Evaluated Providers page.

National Coordinator National Coordinator is a group coordinator responsible for groups that represent countries. Its access to Evaluation was the same as for the Group Coordinator.

4 Pages

The Pending Proposals Page consisted of one dropdown and table element for each time period and one button that triggered the approval of the selected proposals. Only periods in which the coordinator had any proposals left to approve were loaded, and only the table for the opened dropdown was present on the page.

Each row of the table consisted of the name of the provider for which the application calculated the proposal, the evaluation result for each metric, the proposed award, and the number of notes that the coordinators wrote and added to the proposal.

The coordinator responsible for approving the proposals could select multiple rows and approve them by clicking the button. The coordinator could also click on a row in the table, which redirected him to a detail page, in which other information about the clicked proposal was present.

///// This page provided a list of all awards received by the provider to which the Data Entry role was assigned. There was also a button that triggered the recalculation of the latest award ////

The Received Awards Page consisted of the name of the selected provider and a table with all the awards the provider has ever received. The user on this page could filter the awards by time period types and by the fact if the proposal for the award was superseded or not. Time period types were Quarter, Biannual and Annual. They represent the time period in which the providers collected the data needed for the evaluation. I had also to add the current proposal, as it was part of the requirements for this page. Each table row consisted of the period for which the award was given, the evaluation result for each metric, the proposed award, the received award, and the number of notes.

// TODO EVALUATED PROVIDERS PAGE // TODO DETAIL PAGE

When I completed the analysis and understood all the requirements for the final application, I started making prototypes in Figma. I had to make at least three pages. I had to make one page where all pending proposals were listed.

Figma is a collaborative interface design tool. It is widely used in modern application development because it provides a clear user in-

terface and availability to create pixel-perfect designs with the ability to generate CSS styles that a developer can insert into the source code. Another helpful thing Figma provides is the ability to share the work with other people in real time. The most useful thing Figma is capable of is the prototyping ability. I created application-like prototypes. A person viewing the prototype could navigate between different application parts, trigger hover effects, open dropdowns, and click buttons. All of that still would not be functional, and the user actions would not have the desired effect (for example, clicking an Approve button would not approve the selected proposals), but that is not the point of prototypes.

A prototype is an early sample, model, or release of a product created to test a concept or a process. Moreover, that was precisely the goal I needed my prototypes to achieve.