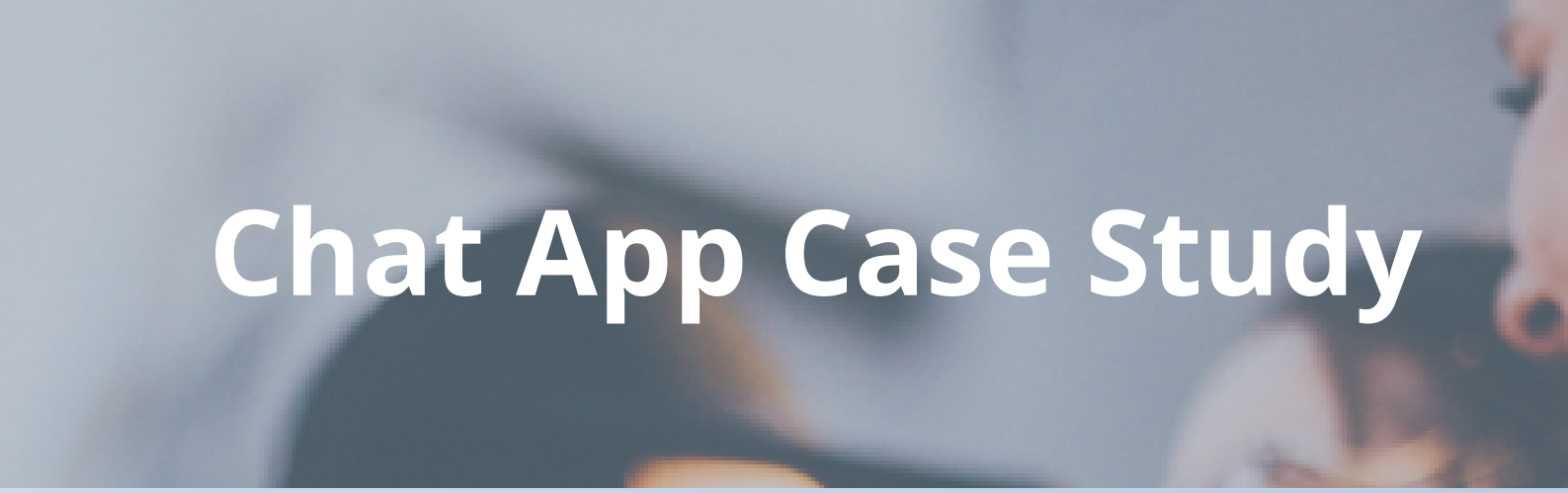


Chat App Case Study

Overview

It is a native real-time chat application, which allows users to talk with each other. The application has only a shared chatroom, where everyone can communicate. Users have the possibility to send images from their phone, take photos and send them, and share their exact location.

Context

It is a mandatory one-person full-stack native app project in CareerFoundry's full-stack web development course.

Objective

To create a chat application, which users can use to communicate with each other, and it is usable with bad or no internet connection, to see the previous messages sent.

Duration

One month. I was able to finish it in time.

Role

Lead developer

Tools

Frontend

- React Native
- Expo
- Android Studio

Backend

- Google Firebase
  - Cloud Firestore
  - Firebase Storage

Developing native applications with React Native

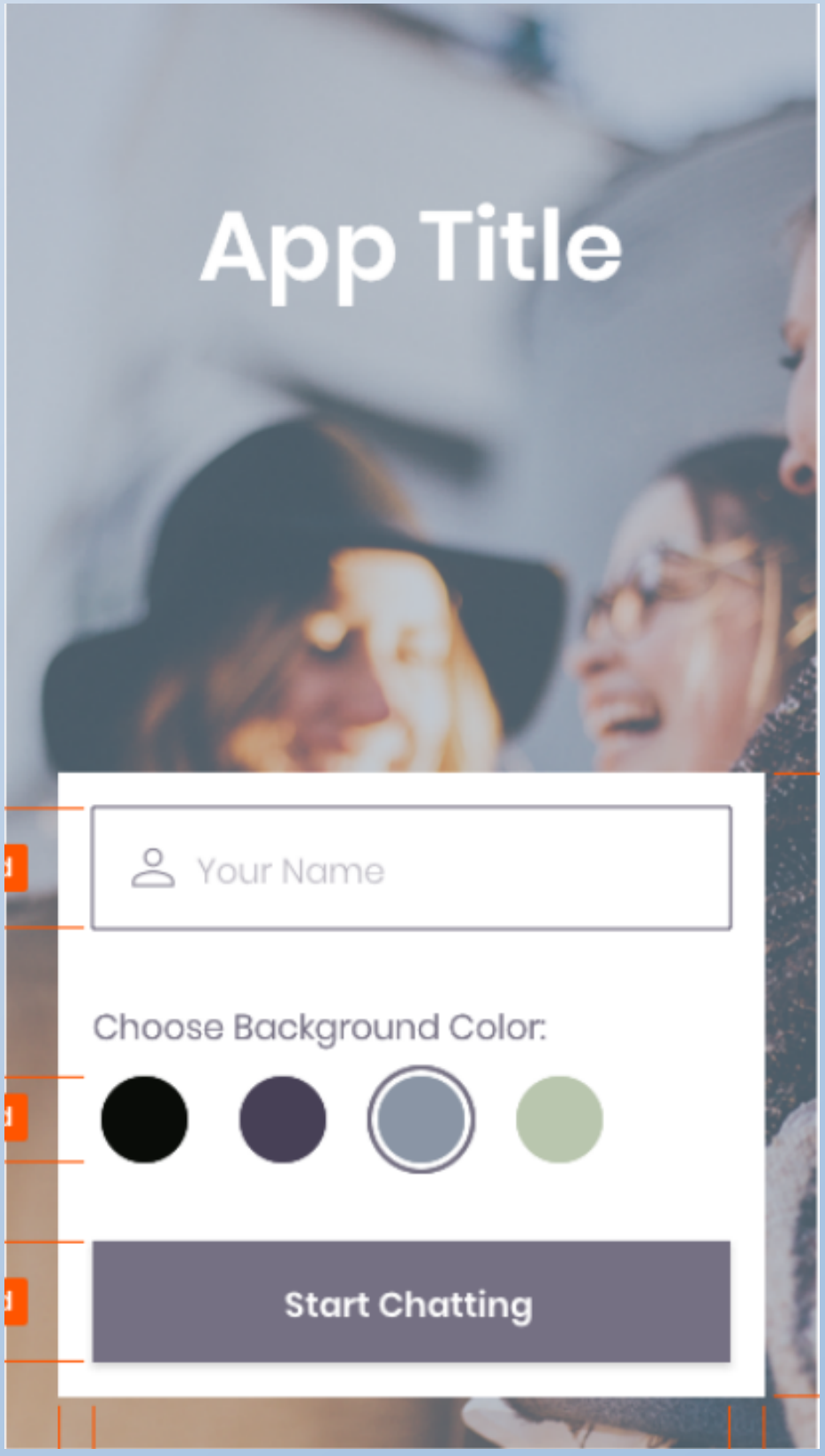
It has a lot of advantages to create a native application through React Native. The greatest advantage is that the application has to be developed only once, and it can be used both on Android and iOS smartphones as well. And a big extra is that an application made with this approach can use the phone's hardware as well, for example, the camera, GPS, and gyroscope.

Everyone who has experience with React can develop native apps easily. It is highly similar to developing a React website.

Creating the screens

I had to make the first screen based on a design. It was less of a challenge after I made the **wedding invitation website**. But it was interesting to know, that with React Native we don't have to specify the unit of measurement, and it will look roughly the

same on every device.



The second screen was a bit of a challenge, since I had to render the background and the bubbles so, that the chosen background color will be used on the second screen, and the bubbles and text have enough contrast to be readable or noticeable for people with vision disability. I used Gifted Chat, which sped up the whole process of the development since it uses a very well-known UI design for chat applications like

Messenger, WhatsApp, and so on. It works somewhat differently from React Native since we have to pass every change as a function to render the look, and every value to a key at the Gifted Chat element. Of that, I felt a bit chaotic to use it at first, but I got used to it fast.

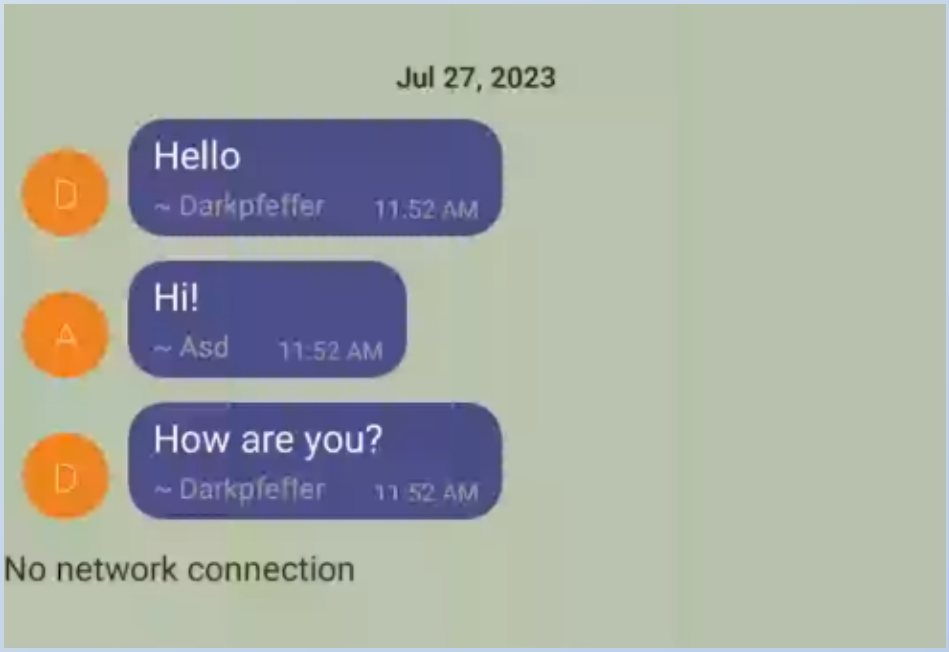
Connecting the application with Firebase

It wasn't a challenge at first to connect my app with Firebase. Everything worked as it should following the documentation. The problem was after I began to use Expo's "Image Picker" library. It worked well until I used it only locally on my phone, but when I connected it to Firebase, it was a real mess. Somehow I couldn't use a basic function called "fetch", because it immediately threw an error. I searched for a solution for about a week until I found the solution on Stack Overflow. I appreciate the solution of "FutureJJ" from **this post.** It was a real lifesaver using 'XMLHttpRequest()' except for 'fetch()'.

Making it possible to use the application offline

For this step, I had to use "React Native Async Storage". This library makes it possible to save data on the user's phone, that way they don't need a stable network connection to use the application and see the previous messages sent and received at the time the user got a network connection. At this step, I didn't have any problem, It fast easy to grasp what is going on. So far, the users can only be logged in anonymously, which means that

they have to open the application before their device lose the connection.



Summary

It felt really fast to develop the application, and I enjoyed the experience of the process. The most challenging part was to be able to upload images to the database of the application. At a second iteration, I would work out a better-looking chat screen, because so far I find it too unpolished, like a low or mid-fidelity prototype.

Video of the application

GitHub page

Thank you for reading through!