Chat mobile app

Development process

September	2023

*All usernames, passwords or other similar information shown in this presentation is fictive and for illustrative purpose only

TABLE OF CONTENTS

TABLE OF CONTENTS

React Native

01

Setting up the development environment and **creating** the project structure, layout and styling

Skills used

Research
Problem-solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

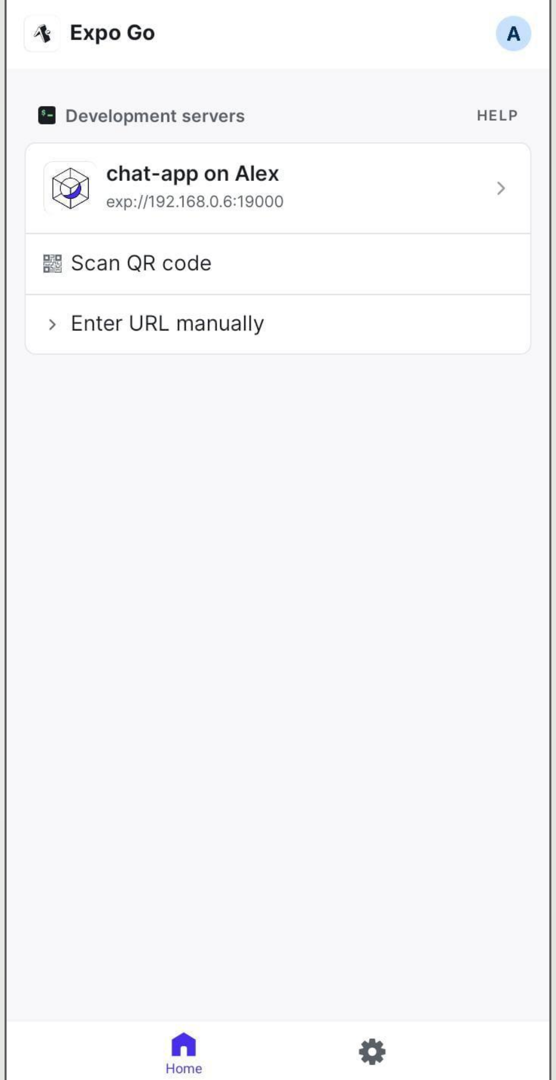
Organizing the development and the coding environment for efficient and well-oriented work from the beginning is one of the keys to ensure smooth workflow and limit avoidable time loss due to inefficient project initialization.

WHAT WAS THE GOAL

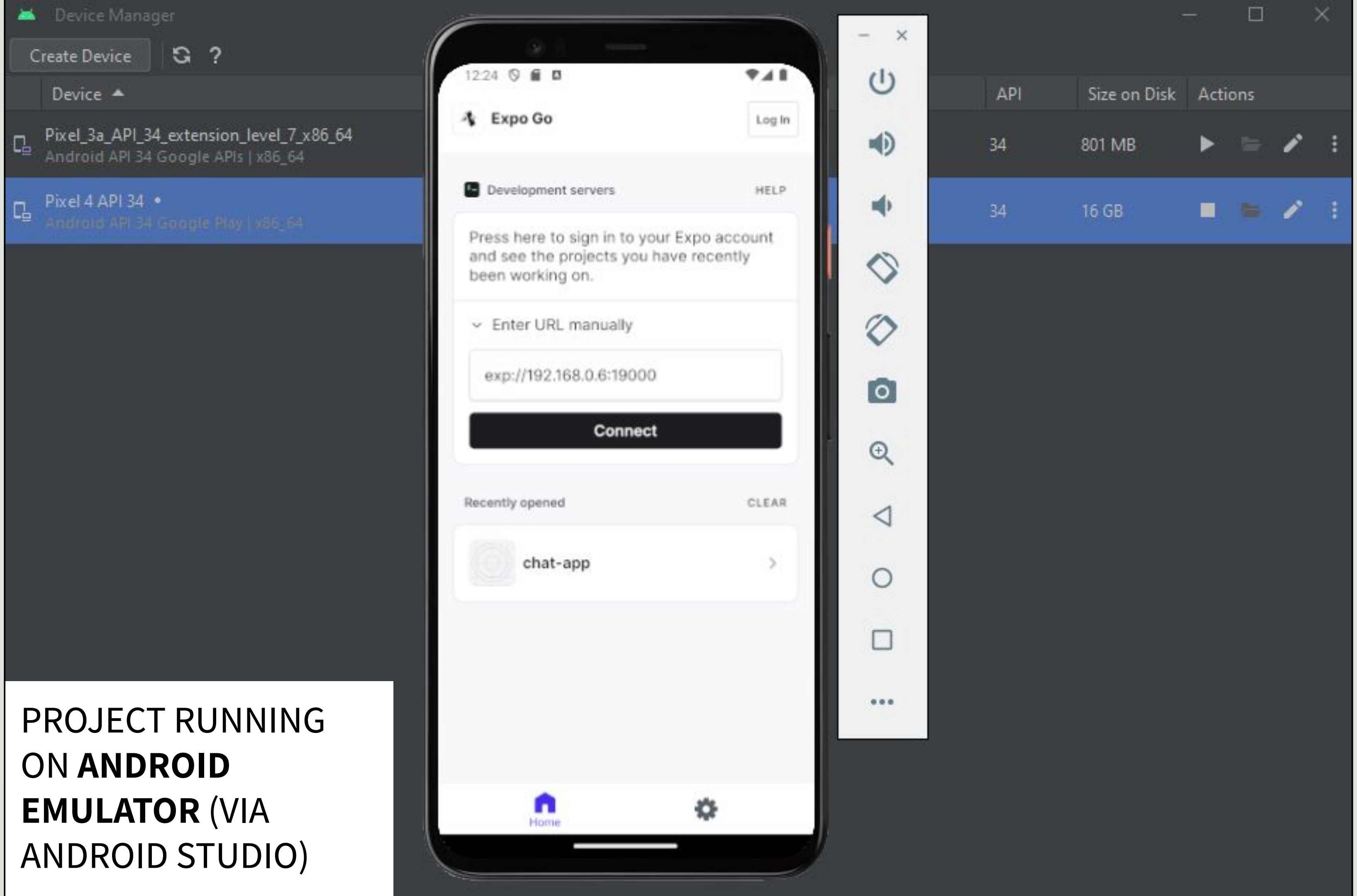
Setting up the development environment to ensure the project is properly initialized. More precisely, I:

- Updated my Node.js to a version supported by Expo;
- Set up Expo and Expo CLI on my machine, as this is the platform used to build the app;
- Set up Expo Go app on my phone, so that it’s possible to test the app on a physical mobile device;
- Set up Android Emulator (Android's virtual device simulation tool) on my machine using Android Studio to see how the app looks and behaves on different devices.

PROJECT RUNNING ON **EXPO GO** ON A PHYSICAL ANDROID DEVICE



chat-app on Alex
exp://192.168.0.6:19000



Device Manager

Create Device

Device	API	Size on Disk	Actions
Pixel_3a_API_34_extension_level_7_x86_64 Android API 34 Google APIs | x86_64	34	801 MB	
Pixel 4 API 34 • Android API 34 Google Play | x86_64	34	16 GB	

12:24

Expo Go

Log In

Development servers

HELP

Press here to sign in to your Expo account and see the projects you have recently been working on.

Enter URL manually

exp://192.168.0.6:19000

Connect

Recently opened

CLEAR

chat-app

Home

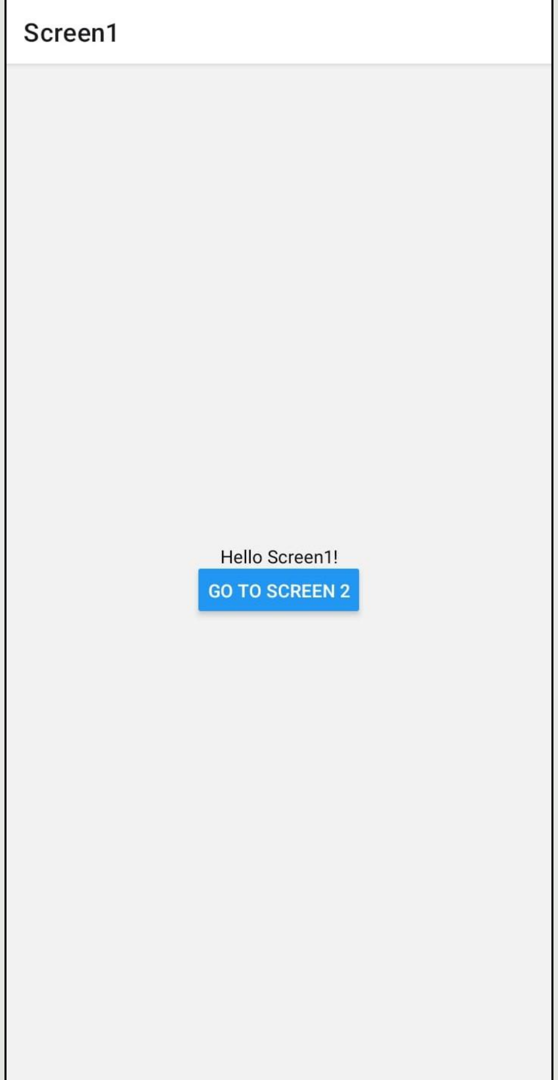
PROJECT RUNNING ON **ANDROID EMULATOR** (VIA ANDROID STUDIO)

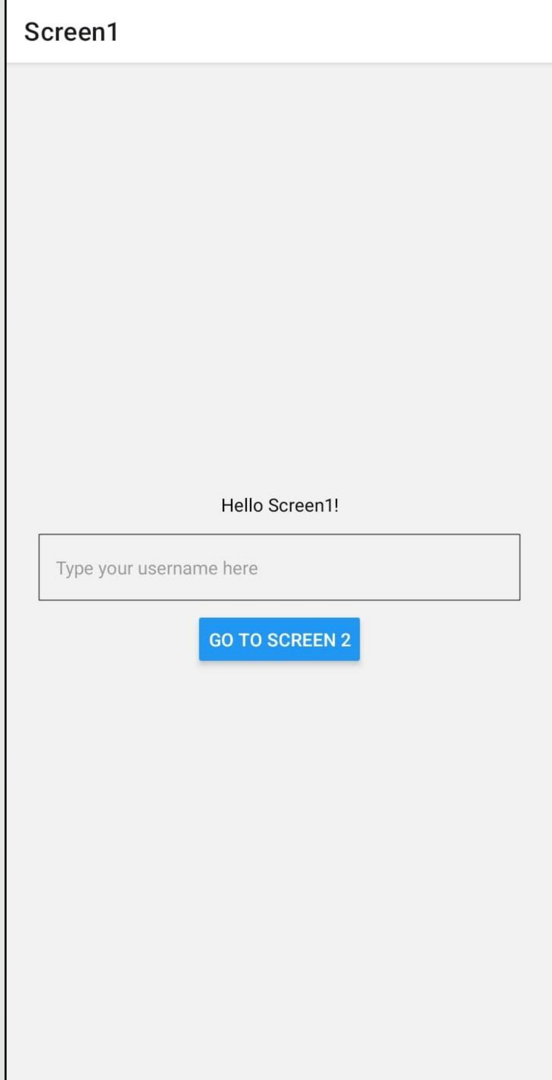
WHAT WAS THE GOAL (SUITE)

Implementing navigation logic between the initial screen and the chat screen (see next images).

Creating the app layout / styling for the initial welcome screen and testing it in my own Android device as well as with Android Emulator from Android Studio (see next images).

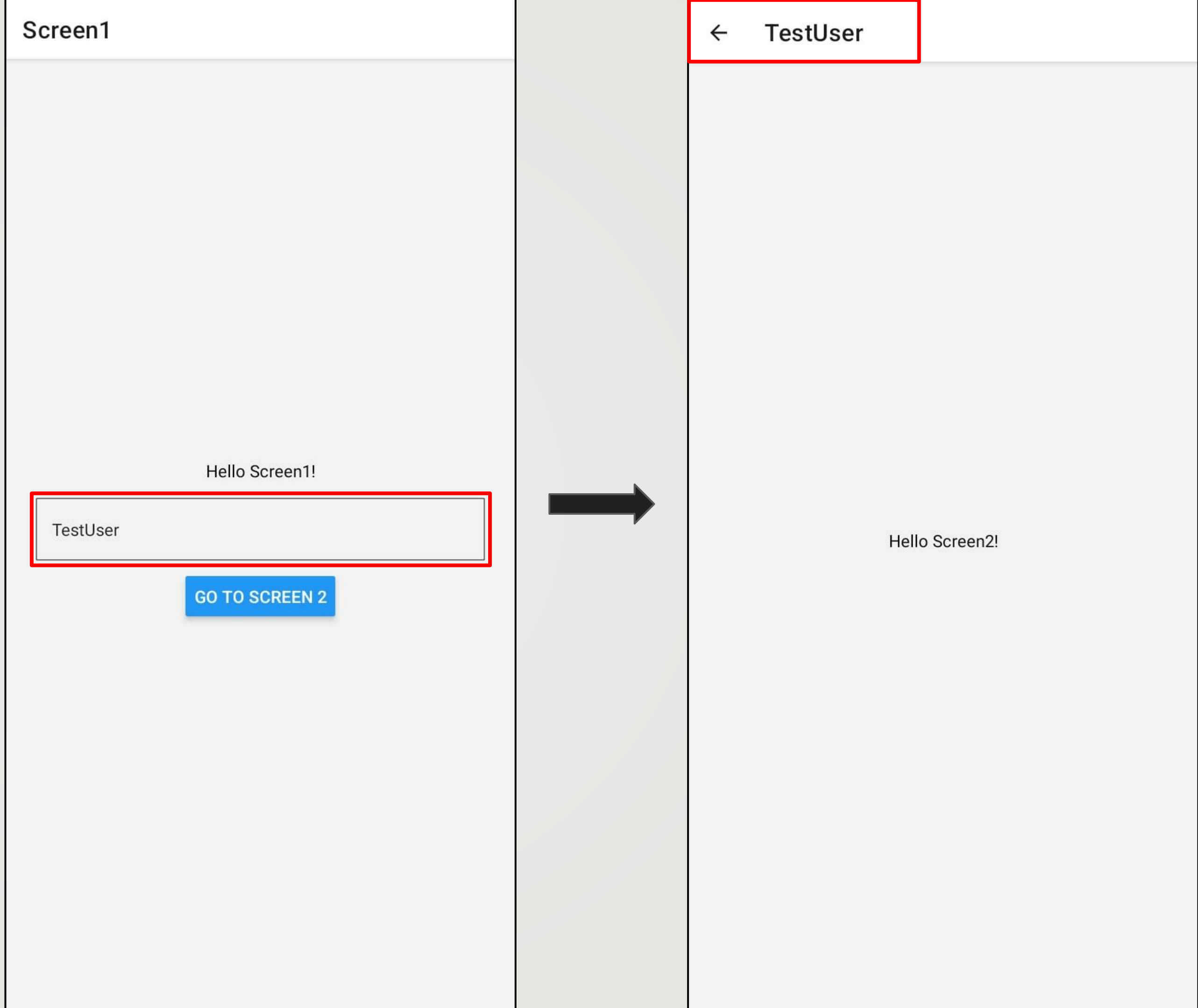
CHAT APP **INITIAL STARTING SCREEN** DEVELOPMENT





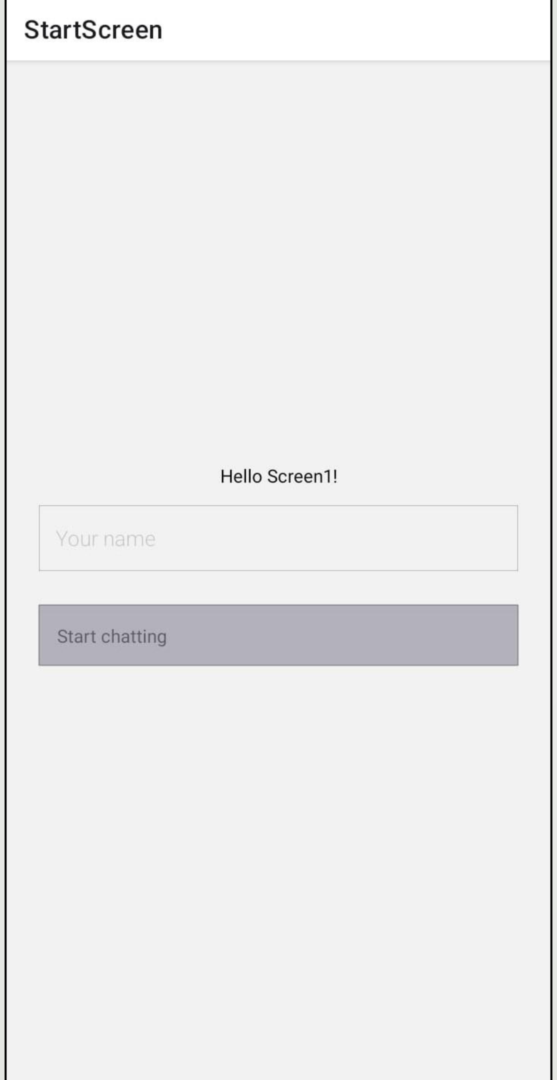
CHAT APP **INITIAL STARTING SCREEN** DEVELOPMENT

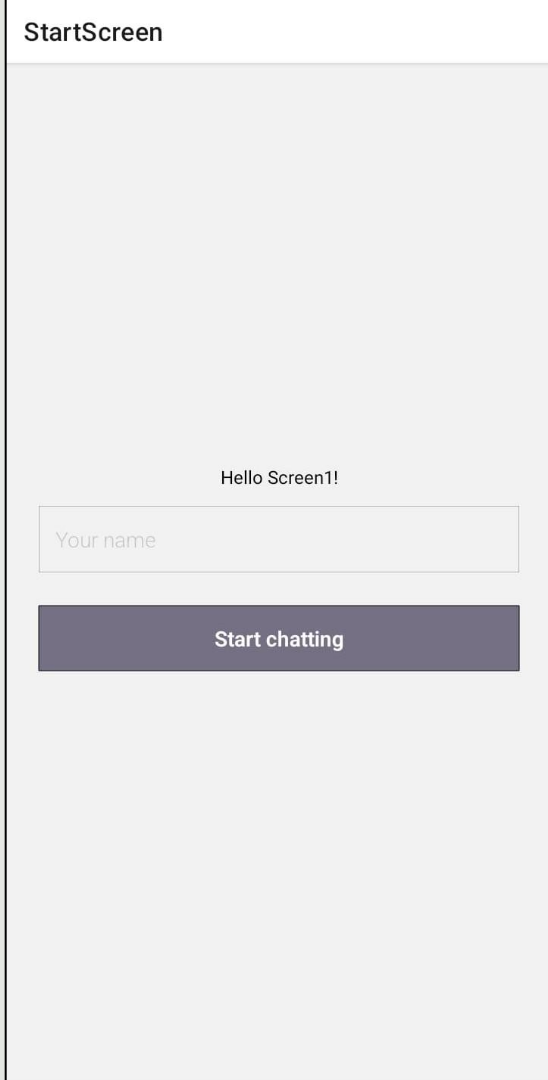
*NAVIGATE FROM THE WELCOME STARTING SCREEN TO THE CHAT SCREEN WHEN THE BLUE BUTTON ON STARTING SCREEN IS PRESSED



CHAT APP **INITIAL STARTING SCREEN** DEVELOPMENT

*STYLING WELCOME STARTING SCREEN ELEMENTS

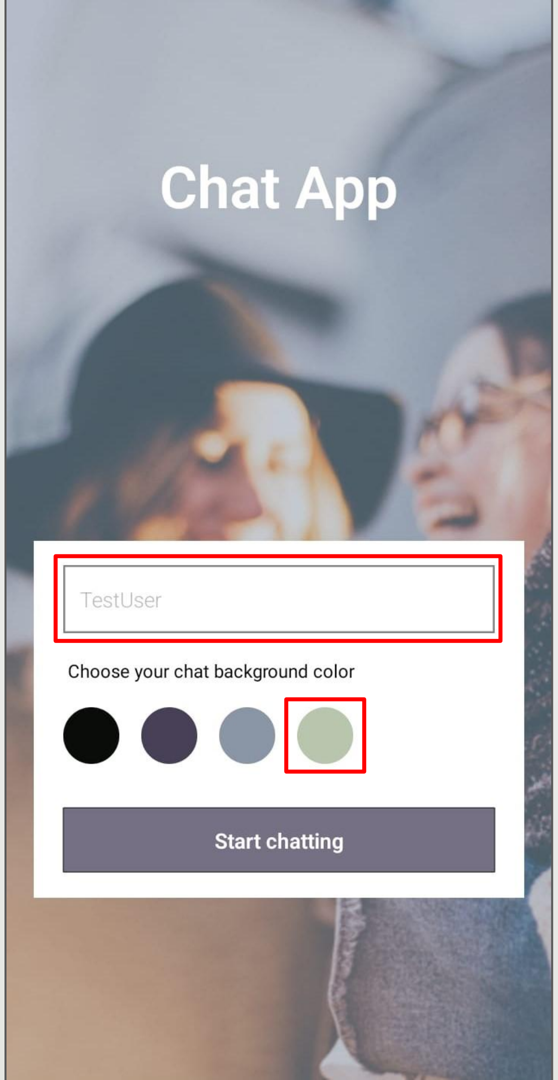


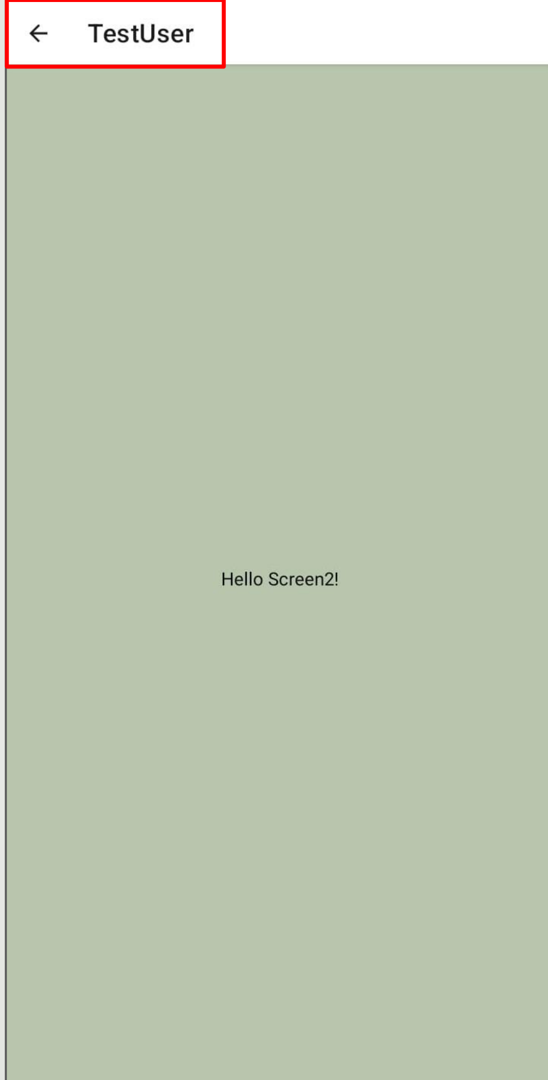


CHAT APP **INITIAL STARTING SCREEN** FINAL LAYOUT

*ALLOW USERS TO ENTER THEIR NAME AND CHOOSE A BACKGROUND COLOR FOR THEIR CHAT SCREEN

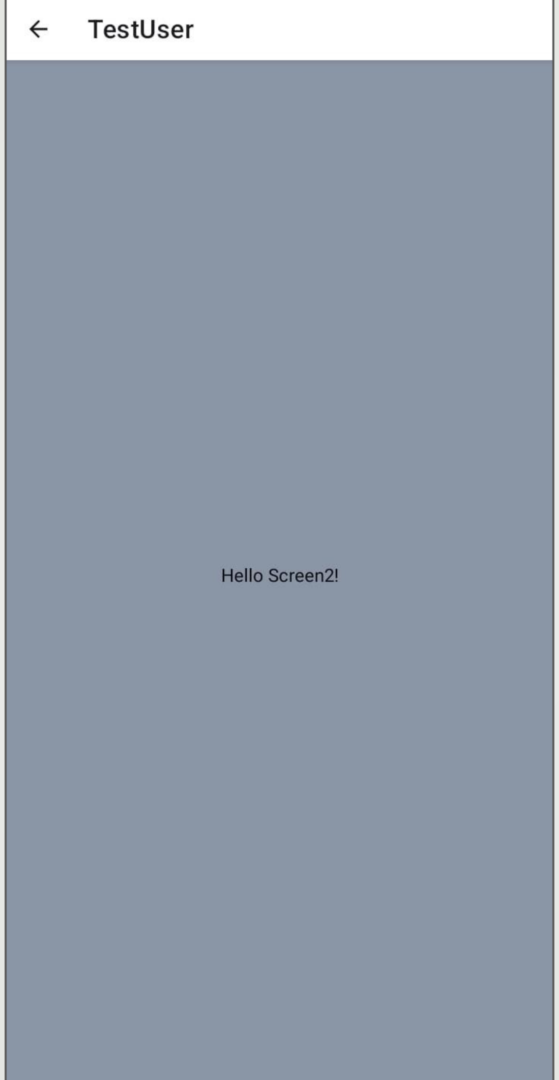
*SECOND IMAGE IS THE CHAT SCREEN, AFTER USERS PRESSED ‘START CHATTING’ ON THE WELCOME STARTING SCREEN



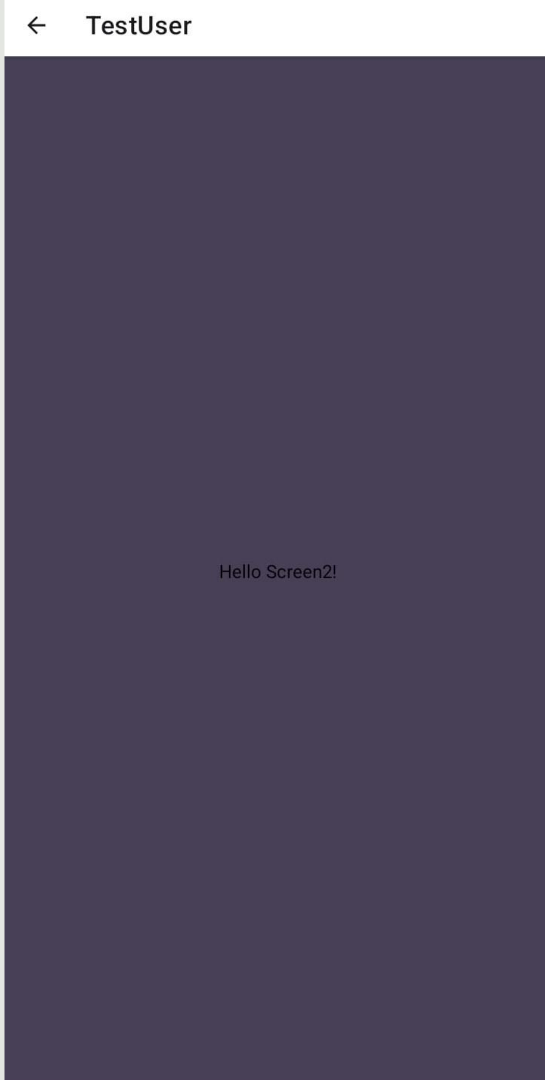


NAVIGATION TO CHAT SCREEN AFTER USERS PRESSED ‘START CHATTING’ ON THE WELCOME STARTING SCREEN

*DIFFERENT BACKGROUND COLORS (MESSAGE COLORS HAVE BEEN ADJUSTED LATER TO ENHANCE VISIBILITY)


TestUser

Hello Screen2!


TestUser

Hello Screen2!

02

Developing the chat screen using Gifted Chat library

Skills used

Research
Problem-solving
Code writing

WHY WAS THIS STEP IMPORTANT

Until this step, the chat screen had no functionalities to send messages. It was therefore important to implement the logic in the chat UI so that users, once on this screen, can send and receive messages in a design that is familiar to them.

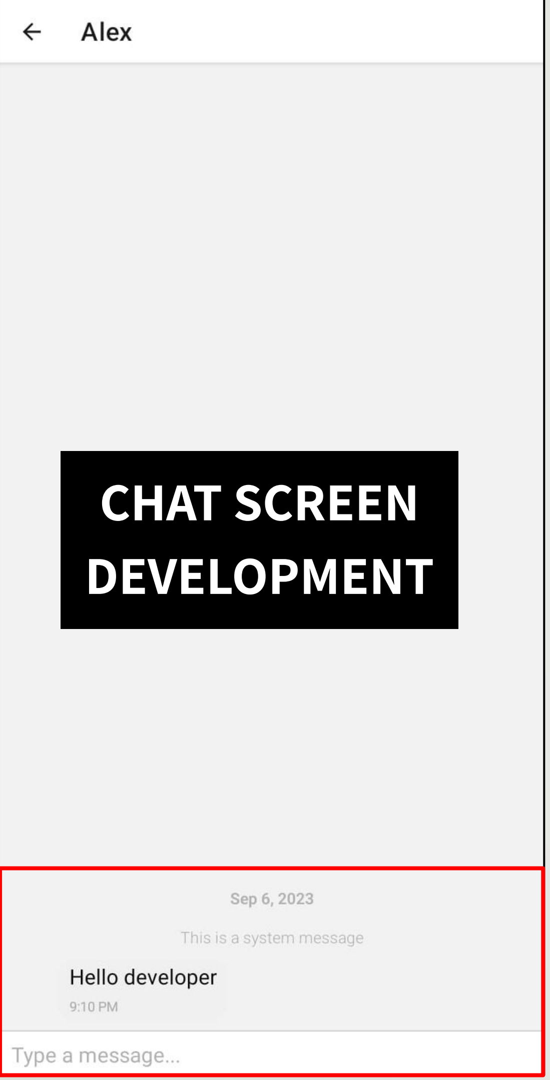
WHAT WAS THE GOAL

Developing the chat screen by using Gifted Chat library to ensure users can send and receive messages (implementation of a text input field for typing messages, speech bubbles for sent and received messages, automatic system messages, etc.).

Styling some aspects of the chat screen (ex: message bubbles) to improve the visual.

Testing everything is working as expected, using both my own Android device with Expo Go and an Android Emulator.

CHAT SCREEN DEVELOPMENT

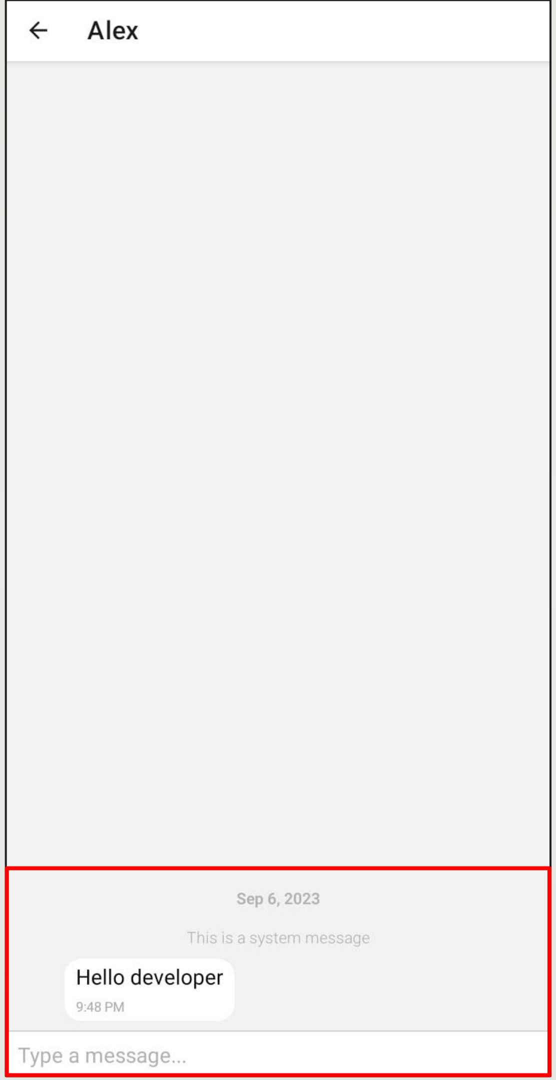

Alex

Sep 6, 2023

This is a system message

Hello developer
9:10 PM

Type a message...

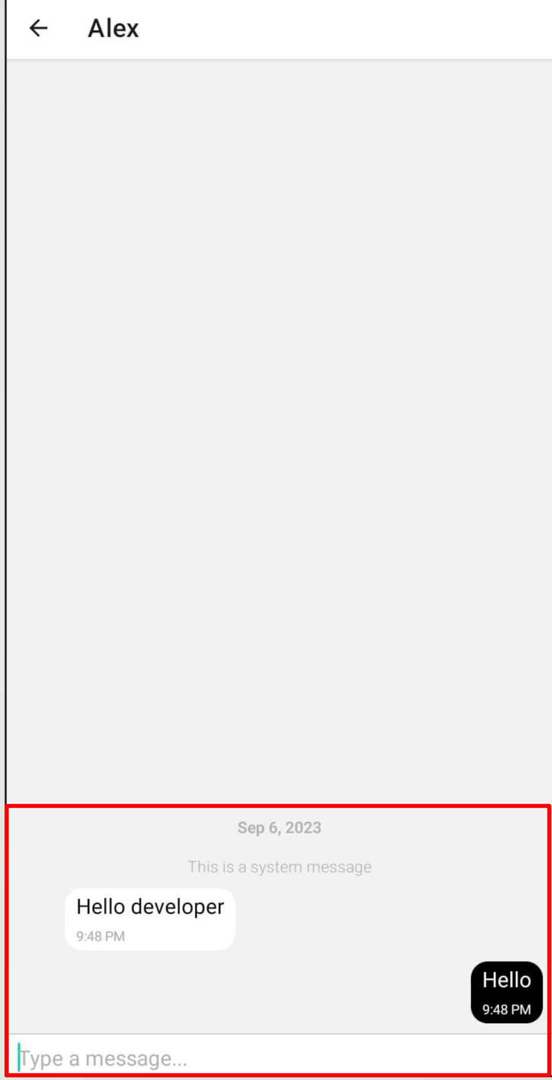

Alex

Sep 6, 2023

This is a system message

Hello developer
9:48 PM

Type a message...


Alex

Sep 6, 2023

This is a system message

Hello developer
9:48 PM

Hello
9:48 PM

Type a message...

03

Connecting the app to a Cloud database

Skills used

Research
Problem-solving
Coding
Debugging

WHY WAS THIS STEP IMPORTANT

This step was important to ensure that the messages sent by users would be stored somewhere and be accessible in real-time. This is needed for a chat app since users expect the messages sent and received to be shown on their screen right away.

WHAT WAS THE GOAL

Setting up a Firestore Database (NoSQL) to save / store the messages on Google Cloud.

- Setting up the Firestore Database was done on the official Firebase website.
- The connection between the Firestore Database and the app was carried out in the app code files, after having installed Firebase as one of the project dependencies and imported all the necessary elements in the project directory.

Implementing codes in the app files so that each message sent is being saved automatically in the Firestore Database, and rendered in real-time in users chat UI.

Cloud Firestore

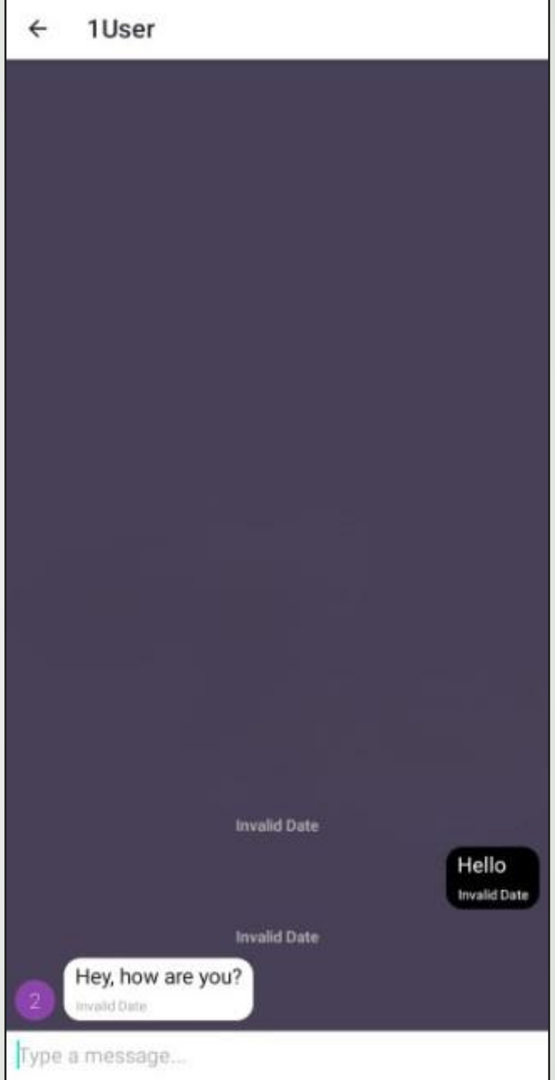
WHAT WAS THE GOAL (SUITE)

Implementing an anonymous authentication process for users using Firebase Authentication feature, to ensure that each user only has access to their messages - and that those messages are not accessible to other users (in other words, ensuring the user experience is personalized).

- Anonymous authentication doesn’t require users to verify their identity by way of an email address (or otherwise).
- Instead, when users sign in in the app for the first time, they are assigned a unique user ID, and this unique user ID is stored both in Firestore Database and locally on their device. Every time they go back on the app, this user ID is retrieved and users get connected with this.

Testing the app to ensure everything works as expected, using both Expo Go on my physical Android device and an Android Emulator.

SAMPLE OF MESSAGES SAVED / STORED **IN THE FIRESTORE DATABASE** AFTER BEING SENT ON CHAT APP BY 1USER AND 2USER



1User

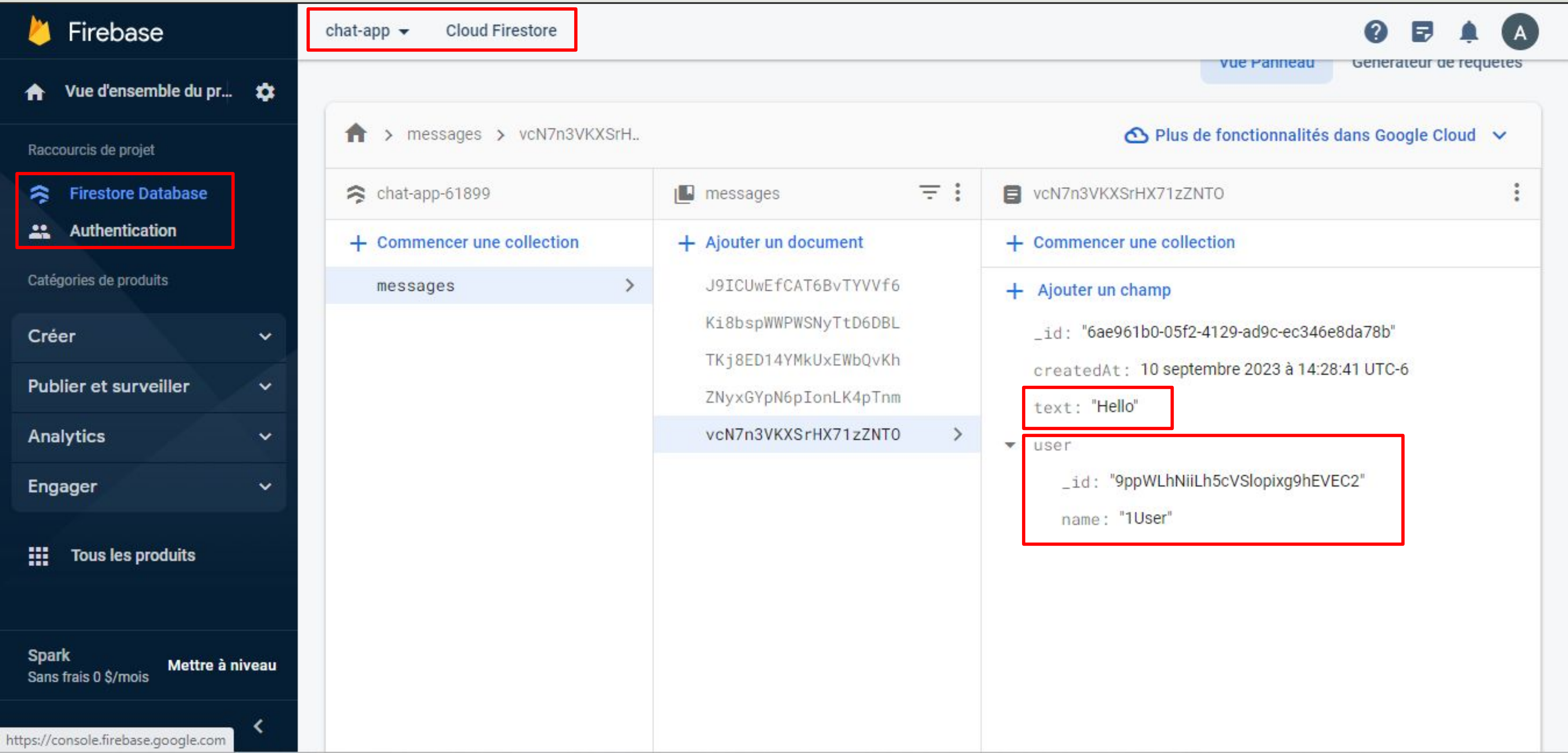
Invalid Date

Hello
Invalid Date

Invalid Date

Hey, how are you?
Invalid Date

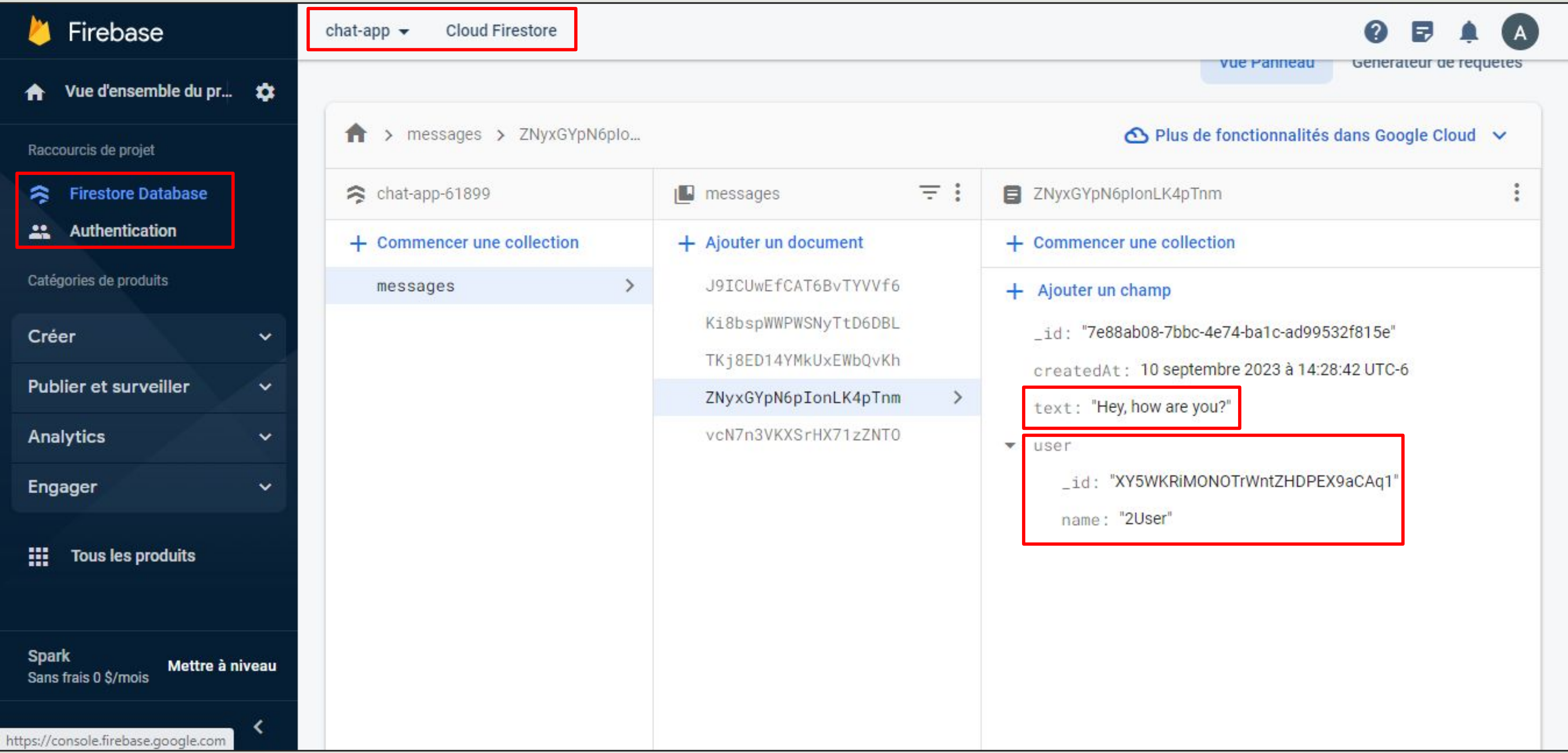
*THANKS TO ANONYMOUS AUTHENTICATION METHOD IMPLEMENTED, EACH USER HAVE THEIR OWN ID



- J9ICUwEfCAT6BvTYVVf6
- Ki8bspWWPWSNyTtD6DBL
- TKj8ED14YMkUxEWbQvKh
- ZNyxGYpN6pIonLK4pTnm
- vcN7n3VKXSrHX71zZNTO

```
_id: "6ae961b0-05f2-4129-ad9c-ec346e8da78b"
createdAt: 10 septembre 2023 à 14:28:41 UTC-6
text: "Hello"
user
    _id: "9ppWLhNiiLh5cVSlopixg9hEVEC2"
    name: "1User"
```

*THANKS TO ANONYMOUS AUTHENTICATION METHOD IMPLEMENTED, EACH USER HAVE THEIR OWN ID



messages
J9ICUwEfCAT6BvTYVVf6
Ki8bspWWPWSNyTtD6DBL
TKj8ED14YMkUxEWbQvKh
ZNyxGYpN6pIonLK4pTnm
vcN7n3VKXSrHX71zZNTO

ZNyxGYpN6pIonLK4pTnm
_id: "7e88ab08-7bbc-4e74-ba1c-ad99532f815e"
createdAt: 10 septembre 2023 à 14:28:42 UTC-6
text: "Hey, how are you?"
user
_id: "XY5WKRiMONOTrWntZHDPEX9aCAq1"
name: "2User"

CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

At first, I tried to initialize my Firestore Database in my code by importing and using the *getFirestore* function from firebase/firestore module. However, this caused incompatibility/network issues. After a few failed attempts to resolve this, I looked for alternatives on Github, and eventually found out that many other developers faced this same problem. Following discussions on the subject, I eventually found a solution and implement it in my code in replacement of the *getFirestore*, which resolves the issues.

04

AsyncStorage

Implementing offline logic and client-side data storage

Skills used

Research
Problem-solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

Users don’t always have an internet connection, so it’s generally very useful to implement some offline logic to make some features available even when there’s no network. This can greatly enhance user experience, especially in a chat type of app.

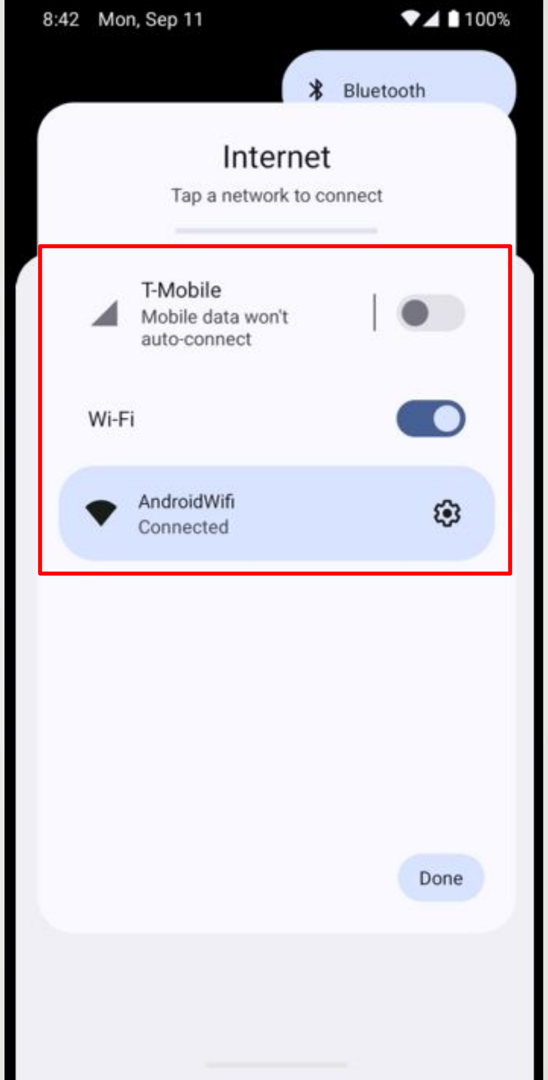
WHAT WAS THE GOAL

Implementing client-side storage in Chat app to allow users to read their messages even when offline. To do so, the package *@react-native-async-storage/async-storage* (*AsyncStorage*) have been used and implemented in the app.

- To determine whether a user is online or not, *NetInfo* have been used (recommended package by the Expo team)
- If there's an internet connection, the data is fetched from the Firestore Database and if there's no internet connection, the data is fetched locally from AsyncStorage. A code has also been written to ensure that the local storage cache is always kept up to date on new messages when there's an internet connection.
- Also, since users can't send messages when they are offline and to make it clear, the text input field is hidden automatically when there's no internet. When the internet connection is back, the text input field is shown again automatically, allowing users to write and send new messages.

Testing the app to ensure everything works as expected, using both Expo Go on my physical Android device and an Android Emulator.

APP RUNNING **WITH INTERNET**


8:42 Mon, Sep 11 100%

Bluetooth

Internet

Tap a network to connect

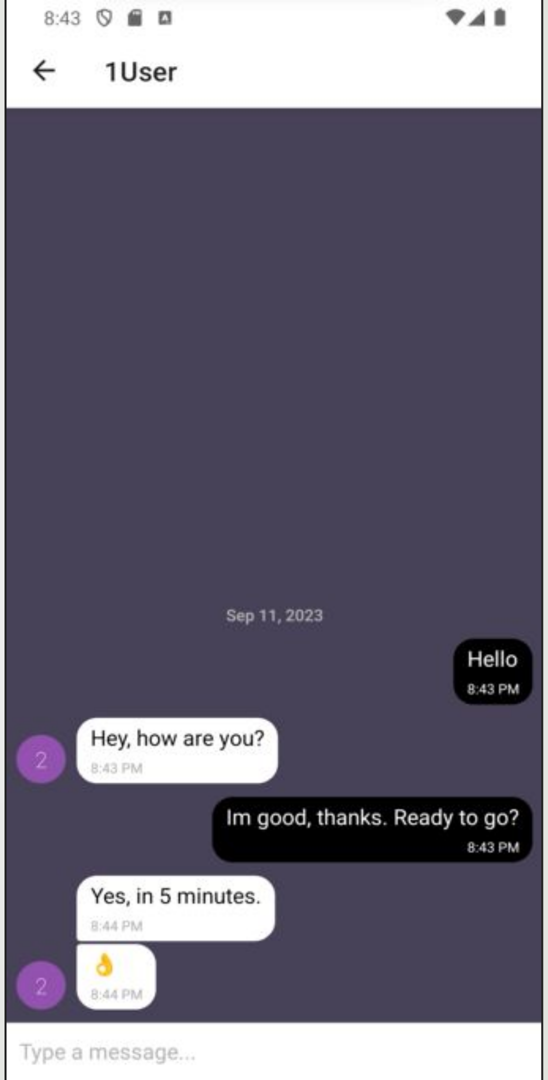
T-Mobile
Mobile data won't auto-connect

Wi-Fi

AndroidWifi
Connected

Done

APP RUNNING WITH INTERNET - **USERS CAN SEND MESSAGES**


8:43

1User

Sep 11, 2023

Hello
8:43 PM

Hey, how are you?
8:43 PM

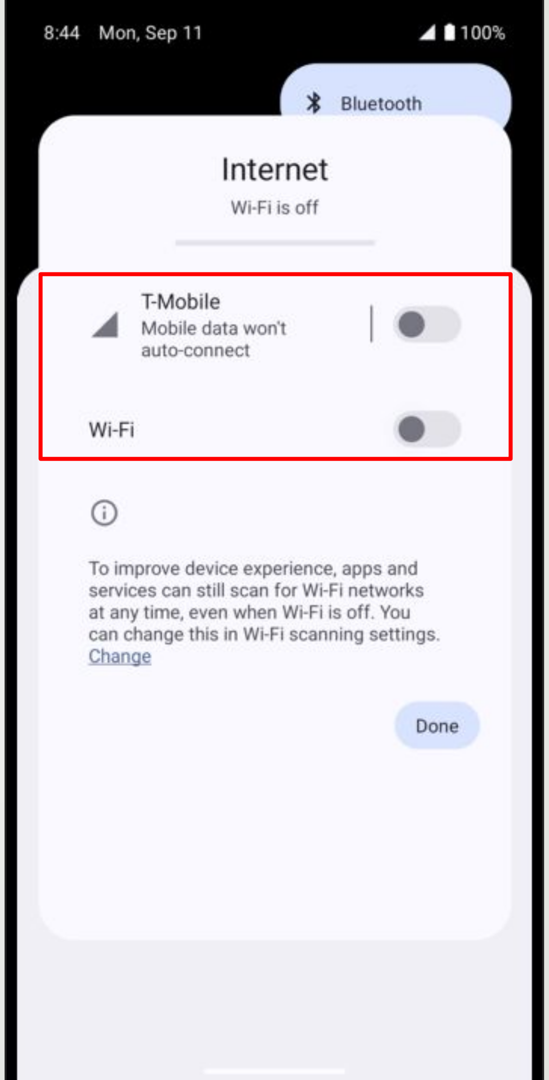
Im good, thanks. Ready to go?
8:43 PM

Yes, in 5 minutes.
8:44 PM

👌
8:44 PM

Type a message...

APP RUNNING **WITHOUT INTERNET**



Internet

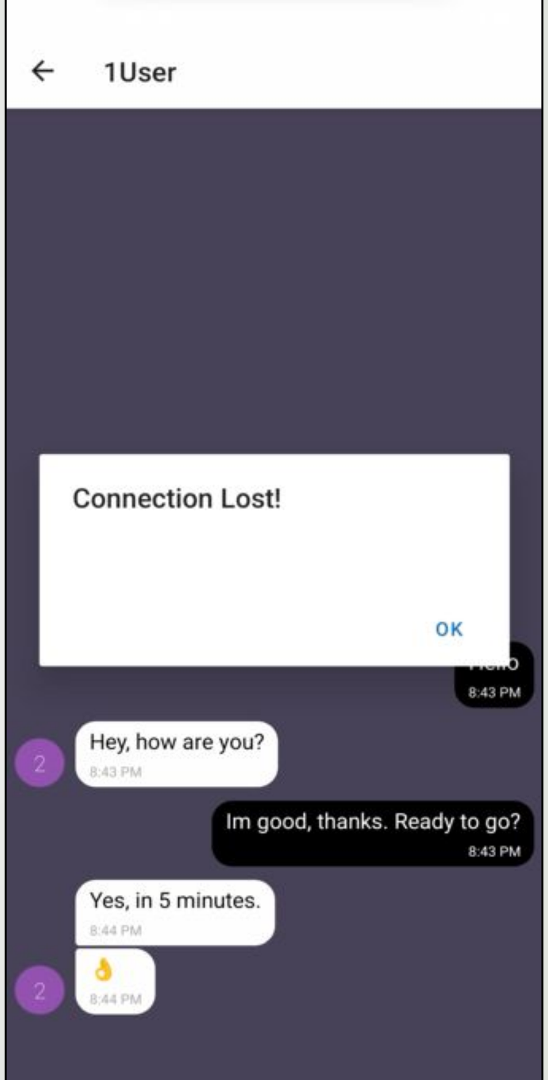
Wi-Fi is off

T-Mobile
Mobile data won't auto-connect

Wi-Fi

To improve device experience, apps and services can still scan for Wi-Fi networks at any time, even when Wi-Fi is off. You can change this in Wi-Fi scanning settings. Change

ALERT LETTING USERS KNOW THEY ARE OFFLINE



Connection Lost!

OK

Hello
8:43 PM

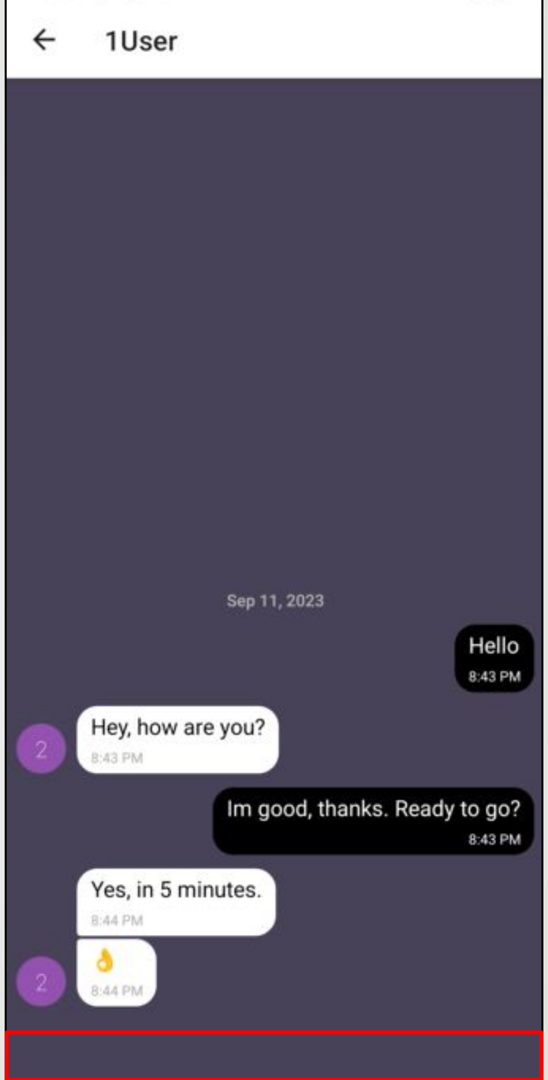
Hey, how are you?
8:43 PM

Im good, thanks. Ready to go?
8:43 PM

Yes, in 5 minutes.
8:44 PM

👌
8:44 PM

APP RUNNING WITHOUT INTERNET - USERS CANNOT SEND MESSAGES (TEXT INPUT FIELD HIDDEN) BUT CAN STILL READ MESSAGES SENT BEFORE


1User

Sep 11, 2023

Hello
8:43 PM

Hey, how are you?
8:43 PM

Im good, thanks. Ready to go?
8:43 PM

Yes, in 5 minutes.
8:44 PM

👌
8:44 PM

05

Implementing communication features and **making the app accessible** for screen readers

Skills used

Research
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

Creating an app that allows users to send messages to each other is great, but for a more complete experience and to ensure to meet the expectations that users generally have for this type of product, it was important to add additional features, such as the ability to send images, take photos, share location and send audios.

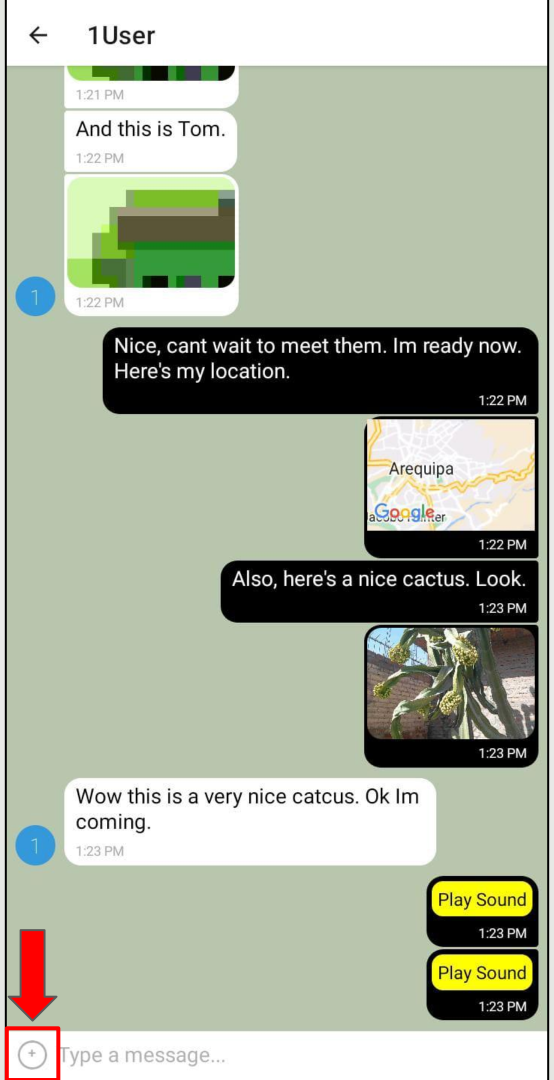
WHAT WAS THE GOAL

Adding additional features in the app to allow users to (1) send images from their phone image library, (2) take pictures with their phone camera and send them, (3) share their location and (4) send audio vocals.

- To allow users to send images from their library or take photos, Expo's *ImagePicker API* has been used.
- To allow users to send their location, the Expo packages *expo-location* and *react-native-maps* have been used.
- Finally, to allow users to send and receive audios, Expo's Audio API (*expo-av*) package has been used.

*Codes have also been added to ensure **users are asked for permission before accessing their library, camera, location and microphone**.

USERS CAN CLICK ON THE **(+) BUTTON** IN THE TEXT INPUT FIELD **TO DISPLAY MORE ACTIONS**



1User

1:21 PM

And this is Tom.
1:22 PM

1:22 PM

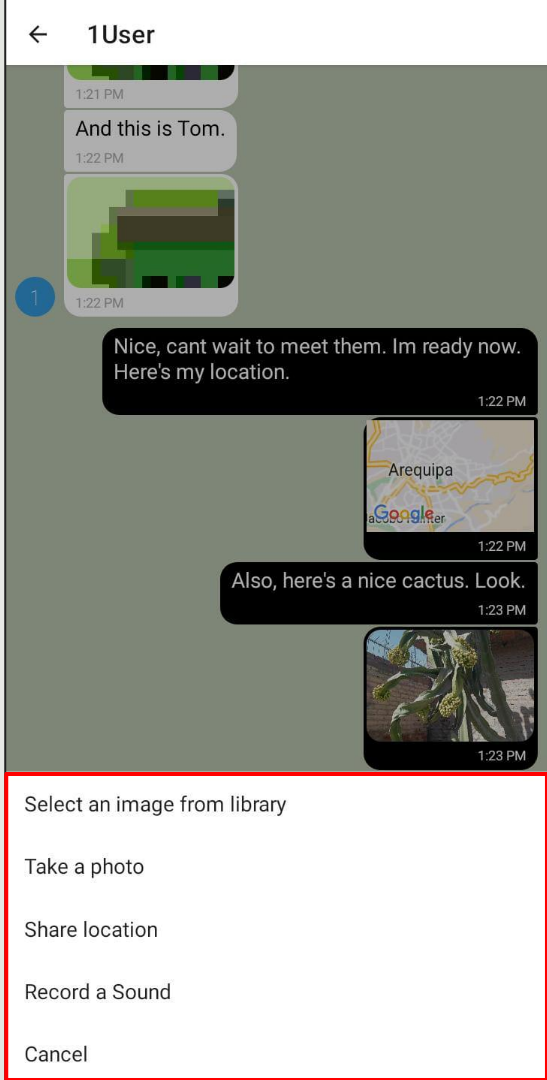
Nice, cant wait to meet them. Im ready now. Here's my location.
1:22 PM

1:22 PM

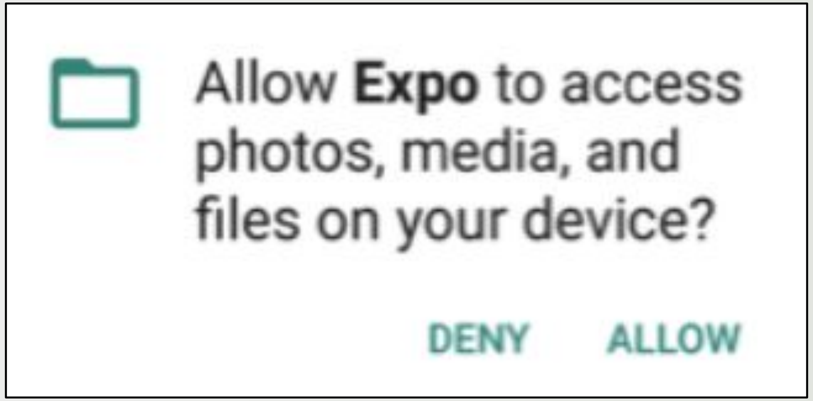
Also, here's a nice cactus. Look.
1:23 PM

1:23 PM

Wow this is a very nice catcus. Ok Im coming.
1:23 PM



PERMISSION ASKED FIRST WHEN USERS CLICK ON ANY ACTION FOR THE FIRST TIME


Allow **Expo** to access photos, media, and files on your device?

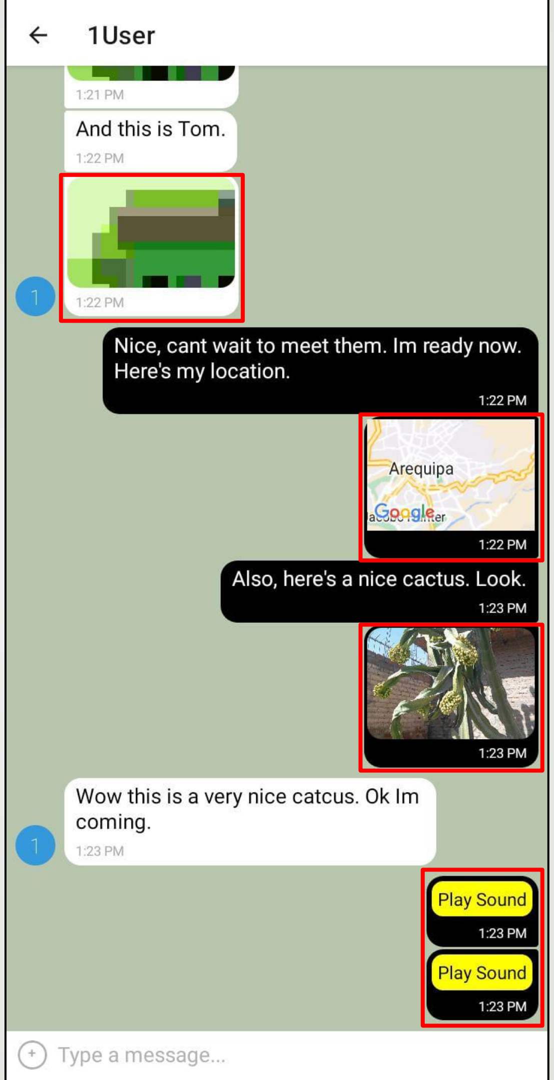
FROM TOP TO BOTTOM:

-IMAGE RECEIVED FROM ANOTHER USER PHONE'S LIBRARY

-SHARED LOCATION

-IMAGE SENT AFTER BEING TAKEN FROM USER PHONE'S CAMERA

-AUDIO VOCALS SENT



1User

1:21 PM

And this is Tom.
1:22 PM

1:22 PM

Nice, cant wait to meet them. Im ready now. Here's my location.
1:22 PM

1:22 PM

Also, here's a nice cactus. Look.
1:23 PM

1:23 PM

Wow this is a very nice catcus. Ok Im coming.
1:23 PM

1:23 PM

1:23 PM

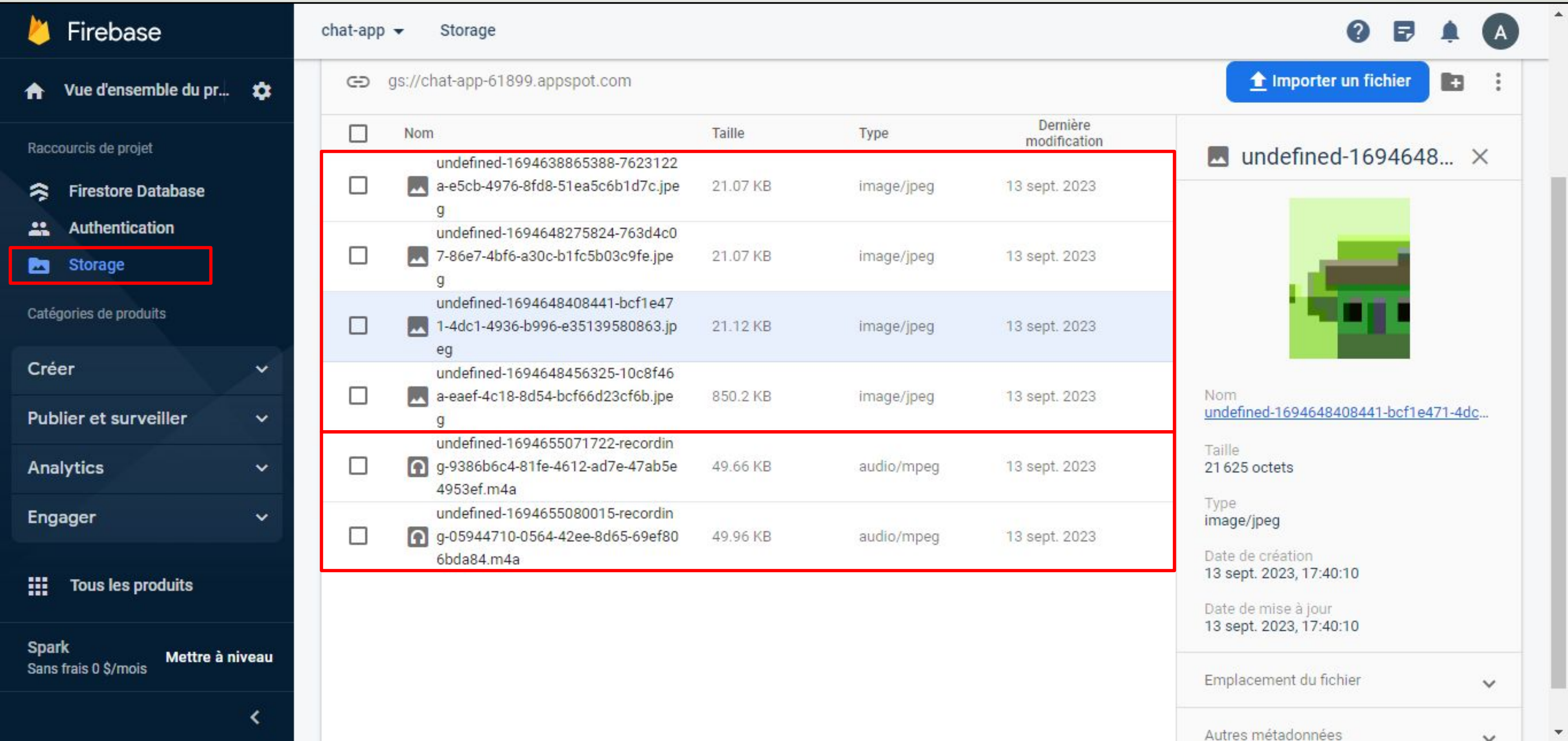
Type a message...

WHAT WAS THE GOAL (SUITE)

Setting up a Google Cloud Storage space on Firebase to store images and audios sent via the Chat app. Storing images and audios on a remote server allow all mobile devices to connect to the server to fetch the sent images and audios, and so access them.

Adjusting the action (+) button (the one users can click on to display more features such as sending pictures from phone's library or share location) to make it more accessible, so that users with screen readers can understand its purpose (addition of an accessibility label, hint and role to the button).

IMAGES AND AUDIOS SENT ON CHAT APP BEING SAVED / STORED IN STORAGE SPACE ON FIREBASE



gs://chat-app-61899.appspot.com

Nom	Taille	Type	Dernière modification
undefined-1694638865388-7623122a-e5cb-4976-8fd8-51ea5c6b1d7c.jpeg	21.07 KB	image/jpeg	13 sept. 2023
undefined-1694648275824-763d4c07-86e7-4bf6-a30c-b1fc5b03c9fe.jpeg	21.07 KB	image/jpeg	13 sept. 2023
undefined-1694648408441-bcf1e471-4dc1-4936-b996-e35139580863.jpeg	21.12 KB	image/jpeg	13 sept. 2023
undefined-1694648456325-10c8f46a-eaef-4c18-8d54-bcf66d23cf6b.jpeg	850.2 KB	image/jpeg	13 sept. 2023
undefined-1694655071722-recording-9386b6c4-81fe-4612-ad7e-47ab5e4953ef.m4a	49.66 KB	audio/mpeg	13 sept. 2023
undefined-1694655080015-recording-05944710-0564-42ee-8d65-69ef806bda84.m4a	49.96 KB	audio/mpeg	13 sept. 2023

undefined-1694648...

Nom
undefined-1694648408441-bcf1e471-4dc...

Taille
21 625 octets

Type
image/jpeg

Date de création
13 sept. 2023, 17:40:10

Date de mise à jour
13 sept. 2023, 17:40:10

Emplacement du fichier

Autres métadonnées

DECISIONS MADE

The audio vocal feature wasn't part of the project requirement. However, I thought it was an addition that could bring great value to the user experience on the Chat app, so I proceed to implement it to enhance available features.

I also added additional code to the anonymous sign in method to ensure that the user ID persists between sessions.

- When users sign in in the app for the first time, a user ID is created, assigned to them and stored locally on their device.
- When users come back on the app, the app re-uses this same user ID to log in users, thus allowing them to access their messages correctly between each session, and where they left it on their last connection.

06

Finalizing code revision, refactoring and last testing

Skills used

Critical thinking

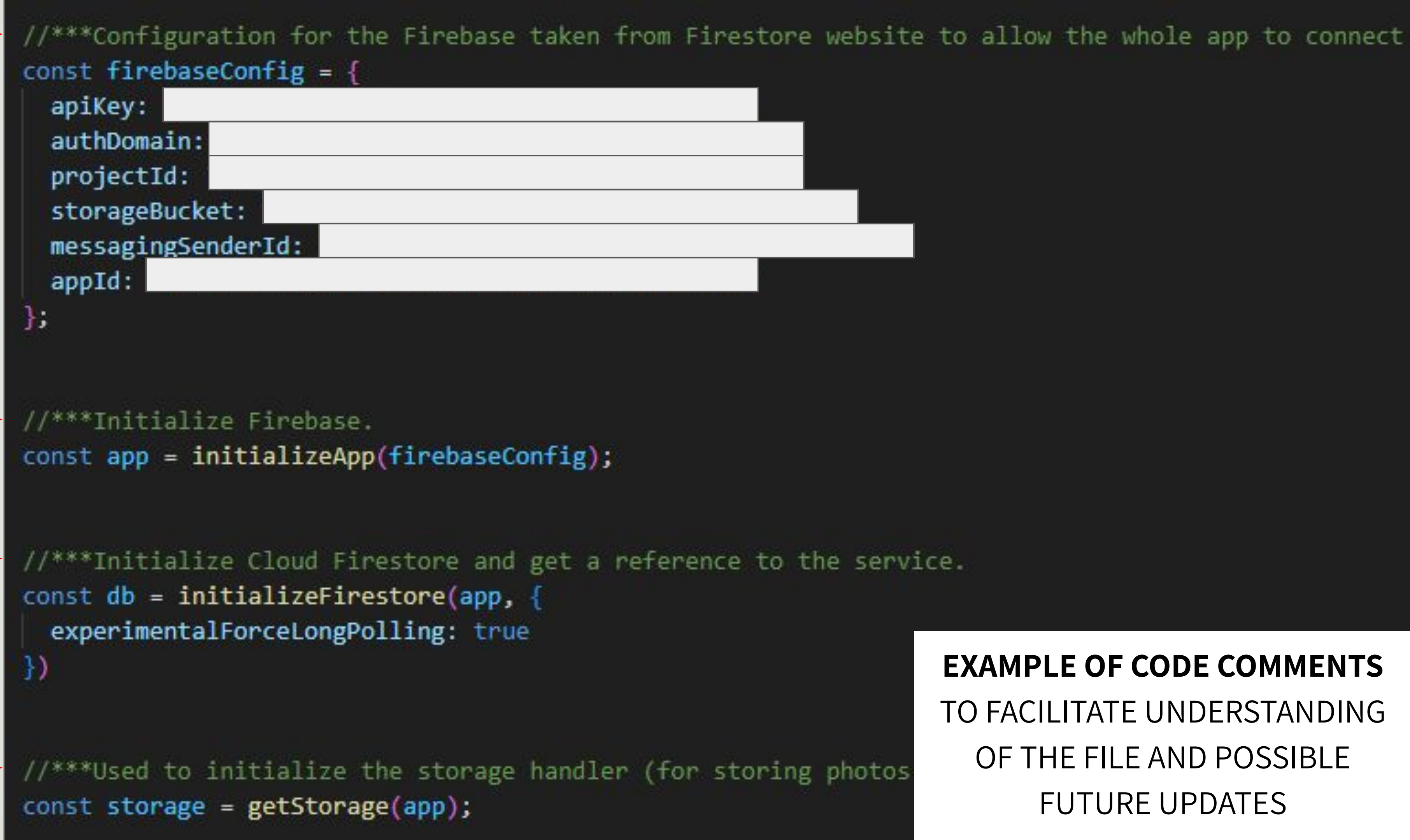
WHY WAS THIS STEP IMPORTANT

Ensuring that the codes are optimized to facilitate possible appropriation by other developers in the future is useful and could possibly save time. It can also facilitate any future adjustments to the codes.

WHAT WAS THE GOAL

Reviewing each code to make sure everything was optimized as much as possible in order to facilitate future modifications, additions or adjustments in the future.

Adding comments and clarification points in the code where important for the benefit and better understanding of anyone else who may work on this project later.



```
//***Configuration for the Firebase taken from Firestore website to allow the whole app to connect
const firebaseConfig = {
  apiKey: [redacted]
  authDomain: [redacted]
  projectId: [redacted]
  storageBucket: [redacted]
  messagingSenderId: [redacted]
  appId: [redacted]
};


//***Initialize Firebase.
const app = initializeApp(firebaseConfig);


//***Initialize Cloud Firestore and get a reference to the service.
const db = initializeFirestore(app, {
  experimentalForceLongPolling: true
})


//***Used to initialize the storage handler (for storing photos
const storage = getStorage(app);
```

EXAMPLE OF CODE COMMENTS
TO FACILITATE UNDERSTANDING
OF THE FILE AND POSSIBLE
FUTURE UPDATES

CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

I positioned myself from the point of view of future colleagues who could work on my project. How can I make my project and my codes as clear as possible to promote its easy appropriation? I reviewed each file to bring improvements in certain places and add comments where I thought it could be useful.

07

README.
md

Completing the README document

Skills used

Communication
Content writing

WHY WAS THIS STEP IMPORTANT

Ensure Meet app is well documented and easily accessible by anyone interested.

WHAT WAS THE GOAL

Updating and completing the README file located in the Meet app Github repository. The goal was to ensure that all relevant information regarding Meet app is accessible under these six categories:

- Project description
- User interface
- User stories
- Technical aspect (overview)
- Technical aspects (development)
- App dependencies

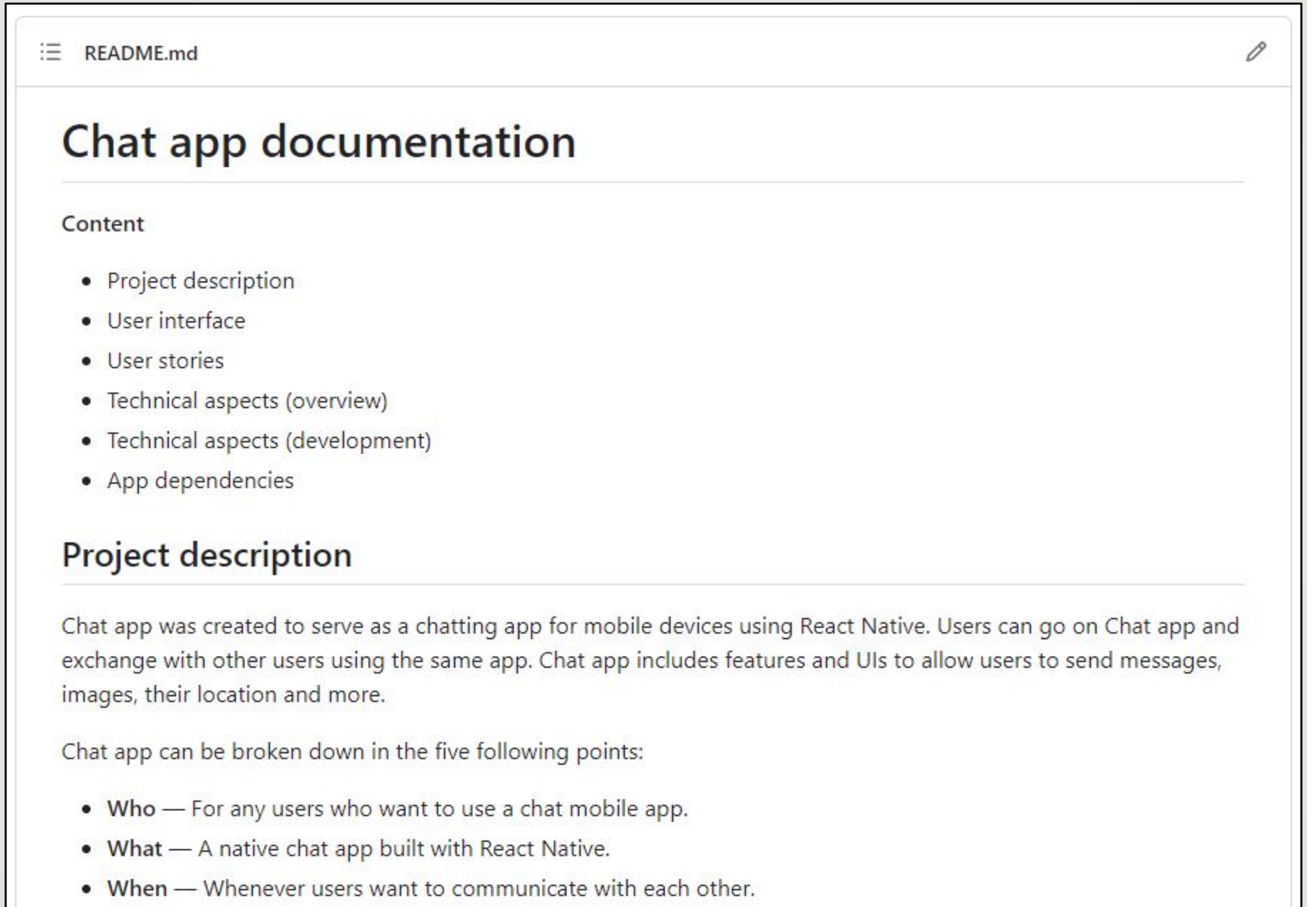
CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

Finding the right balance between giving the right level of information, while remaining as synthetic as possible. To help me, I made a first draft, which I then modified several times. I get inspired by other READMEs I've consulted and for which I found that the information presented was relevant.

DECISIONS MADE

I wrote the README documentation from A to Z, in terms of content, presentation and structure.

README SAMPLE - FULL VERSION ON GITHUB


README.md

Chat app documentation

Content

- Project description
- User interface
- User stories
- Technical aspects (overview)
- Technical aspects (development)
- App dependencies

Project description

Chat app was created to serve as a chatting app for mobile devices using React Native. Users can go on Chat app and exchange with other users using the same app. Chat app includes features and UIs to allow users to send messages, images, their location and more.

Chat app can be broken down in the five following points:

- **Who** — For any users who want to use a chat mobile app.
- **What** — A native chat app built with React Native.
- **When** — Whenever users want to communicate with each other.