

Chatter

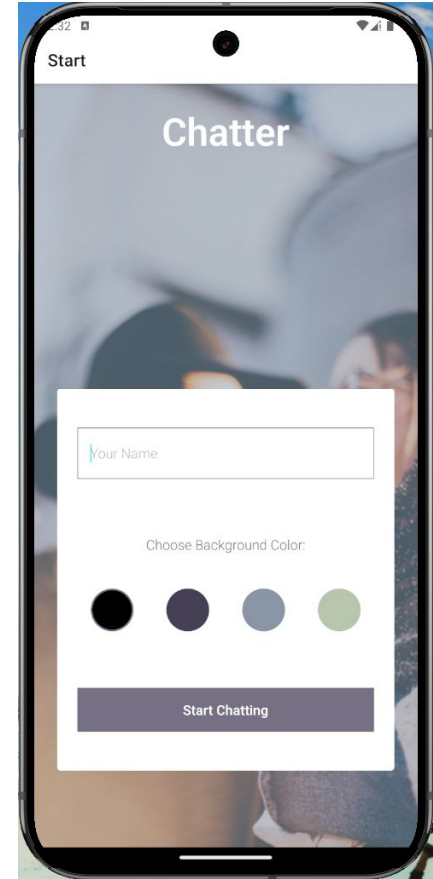
A React Native Case Study
Stephanie Leon

Overview

Chatter is a web application designed using React Native.

Users can join a chat room, choose the background color for the chat screen, and send and receive messages, images and their location.

Messages are able to be read offline and the chat app is compatible with a screen reader.



Purpose & Context



Chatter was created as part of a full stack web development course I completed through CareerFoundry. This was my first opportunity to create a phone application while using React Native and Expo.

Problem Definition

Objective:

The aim of this project was to build a chat app for seamless, real-time communication that supports customization, file sharing, and offline functionality.



Challenges:

- Enable efficient data storage and retrieval
- Ensure the app works across different devices with a consistent user experience
- Incorporate features such as custom actions, media sharing, and offline message handling

Research and Planning

Technology Stack Selection

- React Native for cross-platform development
- Firebase for backend and real-time database capabilities
- Gifted Chat for UI components

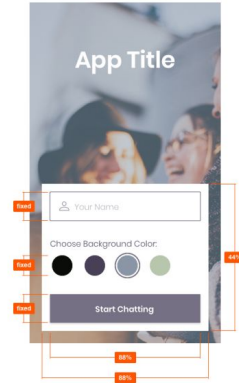
Design Considerations

- User-friendly interfaces
- Scalable architecture for future enhancements

Project Planning

- Define milestones: prototyping, feature development, testing, deployment

Screen Design & Assets



Design Specifications

- Vertical and horizontal spacing: evenly distributed
- App title: font size 45, font weight 600, font color #FFFFFF

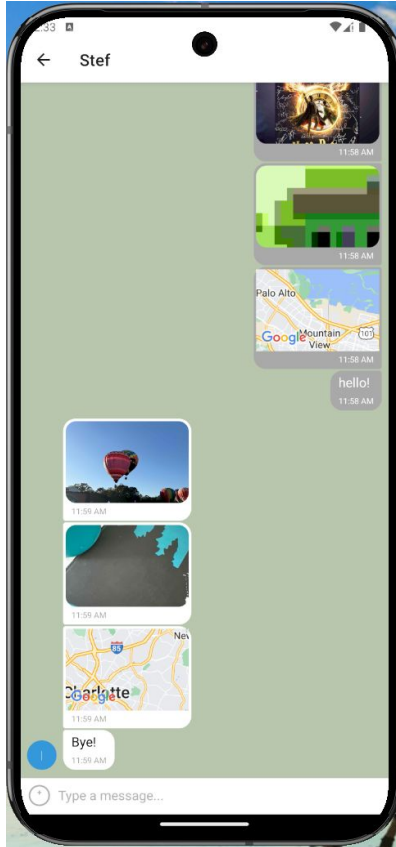
Page 3

- "Your name", font size 16, font weight 300, font color #757083, 50% opacity
- "Choose background color", font size 16, font weight 300, font color #757083, 100% opacity
- Color options HEX codes: #090C08, #474056, #8A95A5, #B9C6AE
- Start chatting button: font size 16, font weight 600, font color #FFFFFF, button color #757083
- [Assets available here](#)

Development Process

Prototyping:

- Basic React Native project setup
- Integration of navigation using React Navigation



Core Development:

- Components
- Features
- Backend Integration
- UI/UX

Development Process Cont.

Components:

- Developed key components (Chat.js, CustomActions.js)

Features:

- Custom actions for sharing and document uploads
- Offline functionality using AsyncStorage

Backend Integration:

- Connected Firebase for real-time messaging
- Defined secure authentication flows

UI/UX:

- Styled chat bubbles and messages lists for intuitive use

Configuration:

- Created configuration files (app.json, package.json)
- Ensured compatibility with various screen sizes and platforms

Testing and Debugging

Unit Testing:

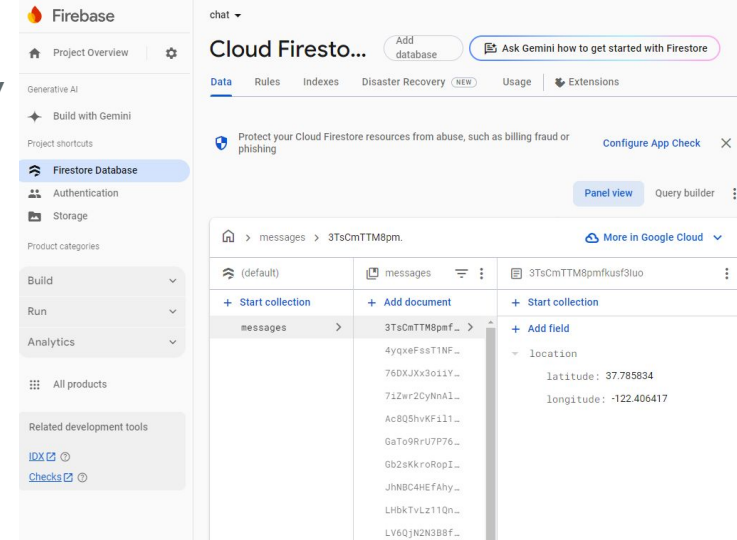
- Tested individual components for functionality
- Validated data flow between components

Integration Testing:

- Verified seamless integration with Firebase
- Ensured custom actions triggered correctly

Performance Testing:

- Assessed app performance under various network conditions
- Optimized for low-memory devices



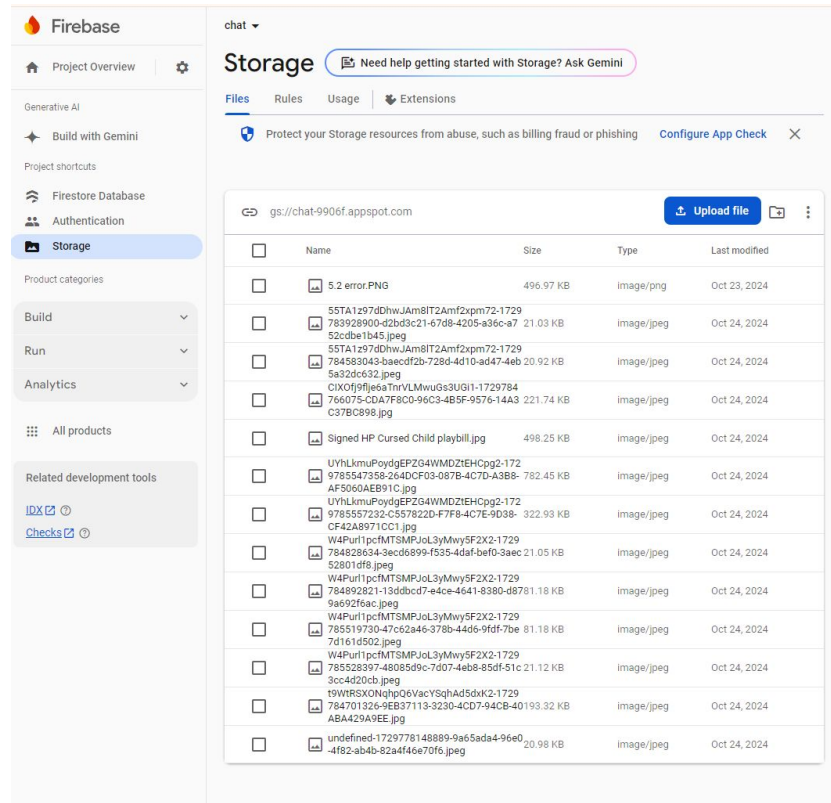
Deployment

Documentation

- Detailed README with instructions for setup, usage, and additional notes for reviewers

Feedback Loop

- Incorporated mentor feedback to improve code quality and user experience



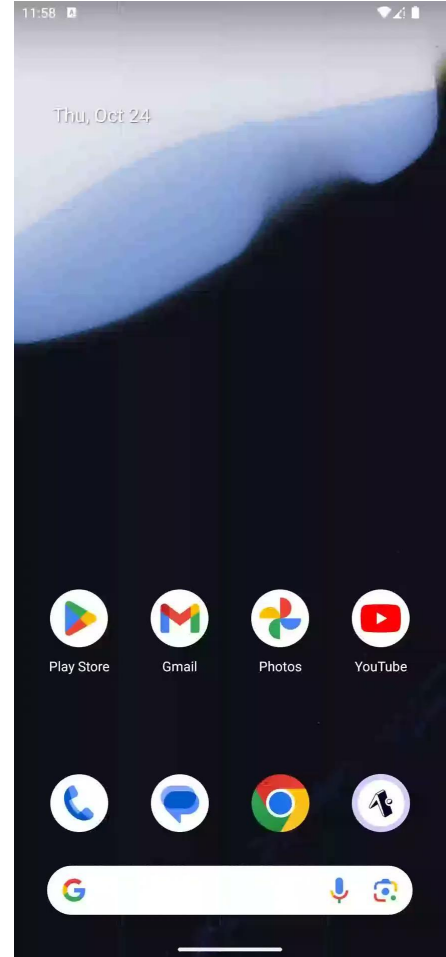
Final Solution

Deliverable:

- A fully functional React Native chat app with the following capabilities:
 - Real-time messaging powered by Firebase
 - Media sharing via custom actions
 - Offline support for seamless user experience
- Scalable architecture for adding new features

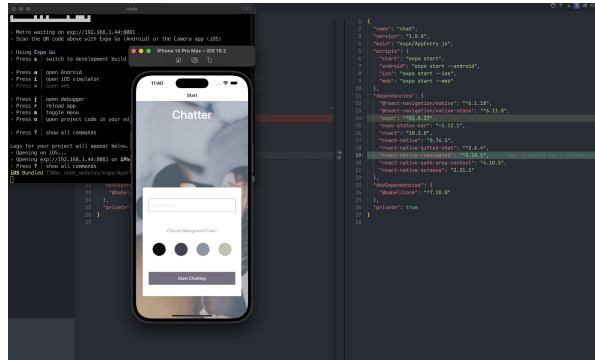
Outcome:

- Successfully launched the app, meeting the defined objectives



Lessons Learned

- Make sure the versions of each package being used are compatible:
 - I ran into issues early in the project with Expo saying it needed to be updated and the fixes provided in the console did not work. I needed to run `npm i expo@latest` and `npm install react-native-reanimated@3.10.1` to fix the issues I was having.
- Using branches in GitHub is helpful when problem solving.
 - I was able to ask for feedback from my tutor, make a new branch and continue working on other features while they reviewed my work.



Future Iterations

- Additional features like recording and playing sounds, uploading an audio file and sending an audio file
- Users able to have multiple chats
- The ability to have a group chat between users

Duration:

I completed the server-side in half a month.

Credits:

Lead Developer: Stephanie Leon

Tutor: Jesus Diaz

Mentor: Shreyansh Kumar