

CAREER**FOUNDRY**

Full-Stack JavaScript Web Development

Achievement 5 Project Brief: Native Mobile Chat App

Objective

To build a chat app for mobile devices using React Native. The app will provide users with a chat interface and the possibility to share images and their location.

Context

More and more people use their phones for daily tasks, such as shopping, creating to-do lists, communicating with friends, scheduling meetings, and more. That's why many companies offer native mobile versions of their web apps, or even skip creating a web app entirely.

In the past, building high-quality mobile apps required a lot of time and money because writing apps for different platforms like iOS and Android required specialized programmers who could build and maintain multiple codebases.

Over time, however, new technologies emerged that made it easier for companies to build and maintain mobile applications using familiar syntax. One of these technologies is React Native, a framework for building Android and iOS apps that only requires one codebase.

For this Achievement's project, you'll build a chat app that you can add to your portfolio to demonstrate your knowledge of JavaScript mobile development using React Native, Expo, and Google Firestore Database.

As you work on this project, you'll use agile project management techniques including user stories, story points, and a kanban board. These techniques are commonly used by companies in the tech industry. The more you know about this topic, the more relaxed it will be for you to settle in with your new work in the future. Additionally, many hiring managers even ask for knowledge or experience with agile from candidates.

Furthermore, you're going to focus on collaboration and documentation, which you'll need in the future when working on teams. The ability to create detailed and straightforward handoffs for other developers is an indispensable skill. It demonstrates an attention to detail as well as the ability to think about others who might work with your code. You'll also learn more about how to work with others by writing and sharing documentation with your peers, providing feedback, and making a contribution to the tech community.

CAREER**FOUNDRY**

The 5 W's

1. Who—The users of the mobile chat app. These could be friends, family or other students on this course. Your codebase will be used by other developers working on the product.
2. What—A native chat app built with React Native, as well as all the relevant documentation. In addition to this, a kanban board containing user stories and story points, and other handoff deliverables.
3. When—Whenever users of your chat app want to communicate with each other. Regarding the documentation, whenever other developers are working on the project.
4. Where—The app will be optimized for both Android and iOS devices. You will use Expo.io to develop the app and Google Firestore to store the chat messages.
5. Why—Mobile chat apps are among the most commonly downloaded and used apps in the world, so knowing how to build a chat app is an indispensable skill. The app will demonstrate your React Native development skills and your ability to create straightforward documentation for other developers and employers.

User Stories

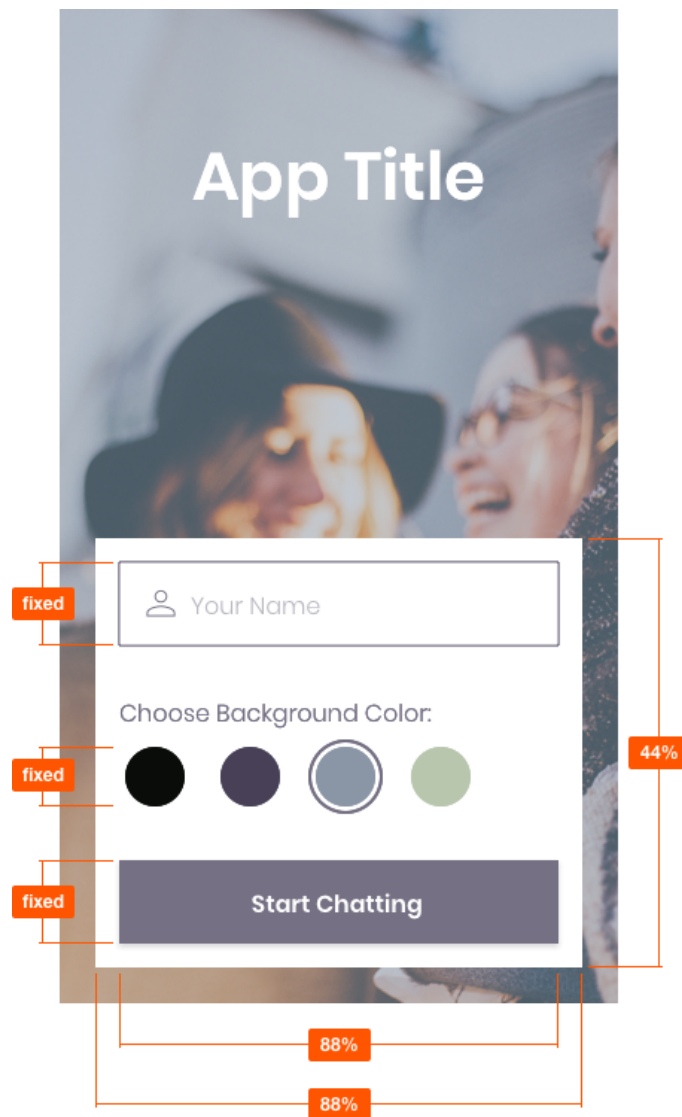
- As a new user, I want to be able to easily enter a chat room so I can quickly start talking to my friends and family.
- As a user, I want to be able to send messages to my friends and family members to exchange the latest news.
- As a user, I want to send images to my friends to show them what I'm currently doing.
- As a user, I want to share my location with my friends to show them where I am.
- As a user, I want to be able to read my messages offline so I can reread conversations at any time.
- As a user with a visual impairment, I want to use a chat app that is compatible with a screen reader so that I can engage with a chat interface.

Key Features

- A page where users can enter their name and choose a background color for the chat screen before joining the chat.
- A page displaying the conversation, as well as an input field and submit button.
- The chat must provide users with two additional communication features: sending images and location data.
- Data gets stored online and offline.

CAREER**FOUNDRY**

Screen Design & Assets



Design Specifications

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

CAREERFOUNDRY

Technical Requirements

- The app must be written in React Native.
- The app must be developed using Expo.io.
- The app must be styled according to the given screen design.
- Chat conversations must be stored in Google Firestore Database.
- The app must authenticate users anonymously via Google Firebase authentication.
- Chat conversations must be stored locally.
- The app must let users pick and send images from the phone's image library.
- The app must let users take pictures with the device's camera app, and send them.
- The app must store images in Firebase Cloud Storage.
- The app must be able to read the user's location data.
- Location data must be sent via the chat in a map view.
- The chat interface and functionality must be created using the Gifted Chat library.
- The app's codebase must contain comments.
- The project must include technical documentation.
- All React Native code must be valid and linted according to given requirements.

Your Project Deliverables

You'll complete your project step-by-step as you work through each Exercise. For each Task, you'll submit a deliverable that directly contributes to the final product—in this case, a native mobile chat app to demonstrate your React and React Native programming skills.

Below is a breakdown of your course project deliverables by Exercise.

Exercise 1: Collaboration and Project Management

- Set up a project kanban board using user stories and story points.
- Create 3 user flows for the chat app.

Throughout this Achievement, you'll be asked to **keep your kanban board up to date**. Make sure to work with your kanban board for each Task, and take some time to reflect on your project planning and assumptions. Were your tasks well broken down and your story points accurate? If not, what could you improve? By doing this, you can practice your estimation skills and get more experienced with agile methods.

Exercise 2: Building Native Applications with JavaScript

- Set up your development environment to work with React Native and Expo.
- Create the app's layout using native UI components and style the start screen based on the screen designs you received in this brief.

CAREER**FOUNDRY**

Exercise 3: Chat UIs & Accessibility

- Build the chat screen and the chat functionality with the Gifted Chat library.

Exercise 4: Real-Time Applications & Data Storage

- Authenticate users anonymously with Firebase.
- Store conversations in the Firestore Database.

Exercise 5: Storing Data on the Client-Side

- Store chats locally using AsyncStorage so they're available offline.
- Authenticate users and store chat messages in Firestore as well as on the device when users are online.
- Retrieve locally stored messages and disallow the creation of new messages when users are offline.

Exercise 6: Communication Features

- Let users pick an image from the device's library and take pictures with the device's camera app.
- Store images in Google Firebase Cloud Storage and send images via Gifted Chat.
- Let users send their current location in a Map view via Gifted Chat.
- Apply accessibility considerations to app design and development.

Exercise 7: Advanced JavaScript & ES6

- Integrate new ES6+ features and advanced JavaScript concepts into your chat app.

Exercise 8: Documentation and Hand-offs

- Native chat app code is commented, documentations is generated, and a project setup manual delivered.
- Final Kanban board is updated (tasks and estimations) and submitted.

Exercise 9: Providing Constructive Feedback

- Review a given screen design and provide constructive feedback.
- Begin portfolio preparation.

Exercise 10: Contributing to the Tech Community & Beyond

- Join online and offline developer communities.
- Contribute to an open source project.
- Incorporate your projects into your portfolio site.