Checker Propriety  
React Native Mobile App

Table of Contents

[The brief: 3](#_Toc130125505)

[Define your app's purpose and target audience: 5](#_Toc130125506)

[Target Audience 5](#_Toc130125507)

[Features: 5](#_Toc130125508)

# The brief:

Creating a proprietary React Native app involves several steps, including planning, setting up the development environment, designing, implementing, testing, and deploying the app. Here's an outline of the process to get you started:

1. Define your app's purpose and target audience:

Begin by identifying your app's primary function and target audience. Determine the essential features you want to include and create a list of requirements.

1. Choose a suitable license:

Since it's a proprietary app, you'll want to choose a suitable commercial license to protect your intellectual property. Consult a legal professional if you're unsure about the type of license to use.

1. Set up the development environment:

Install Node.js, npm (Node Package Manager), and the React Native CLI (Command Line Interface) on your computer. You can find installation instructions on the official React Native website (<https://reactnative.dev/docs/environment-setup>).

1. Initialize your project:

Using the command line, navigate to the directory where you want to create your project and run npx react-native init YourAppName. Replace YourAppName with your desired app name.

1. Design the user interface (UI) and user experience (UX):

Sketch out the app's user interface, keeping in mind the target audience and platform guidelines. Use tools like Figma, Sketch, or Adobe XD to create a design prototype.

1. Develop your app:

Develop your app using the React Native framework, writing components, and implementing features according to your requirements. Use state management libraries like Redux or MobX if needed, and consider using third-party libraries to speed up development.

1. Integrate APIs and services:

If your app requires interaction with external services, integrate relevant APIs or SDKs. This may include authentication, databases, storage, or other third-party services.

1. Test your app:

Test your app thoroughly on both Android and iOS devices to identify and fix any issues. You can use testing frameworks like Jest and Detox, as well as manual testing.

1. Prepare for deployment:

Configure your app for production by optimizing the assets, updating the app icon and splash screen, and ensuring that the app's metadata (name, version, etc.) is correct.

1. Deploy your app:

Deploy your app to the Google Play Store and Apple App Store by following their respective guidelines. You'll need to create a developer account for each platform, submit your app for review, and manage its release.

1. Provide ongoing support and updates:

After launching your app, ensure that you provide regular updates and address user feedback. Monitor your app's performance, fix bugs, and add new features as necessary.

# Define your app's purpose and target audience:

Target Audience:

1. Geographic location: The target audience is primarily people living in the United Kingdom. This includes major cities, towns, and rural areas throughout the country.
2. Age group: The app targets users across a wide age range, from young adults (18-24 years) looking for their first property to rent or buy, to older adults (55+ years) considering downsizing or relocating.
3. Income level: Users with varying income levels will use the app, including those searching for affordable housing, middle-income earners, and high-income earners looking for luxury properties.
4. Life stage: The target audience includes individuals at different stages in their lives, such as students, single professionals, couples, families with children, and retirees.
5. Occupation: Users from various professional backgrounds and industries will use the app, including employees, self-employed individuals, business owners, and investors.
6. Property needs: The app caters to users with diverse property needs, such as first-time buyers, renters, property investors, and those looking to sell or lease their properties.
7. Interests and preferences: The app targets users with various interests and preferences related to property search, including eco-friendly or sustainable housing, properties near specific amenities (e.g., schools, parks, or public transportation), and those seeking particular architectural styles.

## Features:

1. User authentication (required): Allow users to create an account and sign in to save their preferences, search history, and favorite properties.
2. Property search: Implement a powerful search functionality that allows users to find properties based on various filters such as location, property type, price range, number of bedrooms, and other customizable criteria.
3. Property details: Display detailed information about each property, including photos, descriptions, floor plans, features, and contact information for the property owner or agent.
4. Map integration: Integrate a map service like Google Maps or Mapbox to display properties on a map and provide location-based search options.
5. Favorite properties: Allow users to bookmark their favorite properties and easily access them later.
6. User profile and preferences: Provide a user profile section where users can manage their personal information, preferences, and search history.
7. Contact options: Include an in-app messaging system or a contact form for users to directly communicate with property owners or agents.
8. Ratings and reviews: Offer a feature for users to rate and review properties, helping others make informed decisions.
9. Responsive design: Ensure that the app is compatible with both Android and iOS devices and provides an optimal user experience on various screen sizes.

# UI:

1. Design Software: Use design software like Adobe Illustrator, Sketch, Figma, or an online tool like Canva to create your logo. These tools provide the necessary features to create vector graphics and export them in various formats.
2. Typography: Choose a modern and readable font for your app's name or initials that complements the logo design. Ensure that the font is easily legible, even at smaller sizes.
3. Export and Use: Export your logo in various formats, such as PNG, JPEG, or SVG, and use it within your app, marketing materials, and other brand assets.

## Color Scheme:

Since your target audience is broad, including anyone who is interested in checking properties, you should aim for a color palette that is universally appealing, clean, and user-friendly. Here's a suggested color scheme for your property checker app:

1. Primary color: A shade of blue, such as #4285F4, which represents trust, stability, and professionalism. Blue is a versatile color that works well for a wide range of audiences.
2. Secondary color: A shade of green, such as #49A342, which symbolizes growth and prosperity. Green complements blue and is associated with positive emotions.
3. Accent color: A shade of orange or yellow, such as #FFC82F, for attention-grabbing elements like buttons, icons, and progress bars. This color adds warmth and energy to the overall design.
4. Neutral colors: Light and dark shades of grey for text, backgrounds, and borders, such as #F5F5F5 for a light background and #333333 for dark text. These colors ensure readability and visual balance.

To ensure that your app appeals to a broad audience, focus on creating a clean and intuitive user interface, with easy-to-read text, clear navigation, and a responsive design that works well on various devices and screen sizes. By incorporating a color scheme that is widely appealing and user-friendly, your property checker app will be more likely to resonate with your target audience.

Components:  
 0. iOS Design Kit: This kit contains a comprehensive collection of components, templates, and UI elements designed specifically for iOS apps. It follows Apple's Human Interface Guidelines and includes components for both iPhone and iPad. Figma community link: <https://www.figma.com/community/file/733673533348576960>

1. Material Design Kit: This kit is based on Google's Material Design guidelines and includes a wide range of components, templates, and UI elements tailored for Android apps. Figma community link: <https://www.figma.com/community/file/778763161265841811>
2. Ant Design Mobile UI Kit: This UI kit is based on the Ant Design mobile component library, which is popular for creating responsive and modern mobile app designs. Figma community link: <https://www.figma.com/community/file/908389479886731720>
3. Fluent Design UI Kit: This kit is inspired by Microsoft's Fluent Design System and provides a wide variety of UI components suitable for designing mobile apps with a modern and clean aesthetic. Figma community link: <https://www.figma.com/community/file/830240416716111597>
4. Figma Mobile UI Kit: This is a versatile mobile UI kit that includes a variety of components and UI elements for designing mobile apps across different platforms. Figma community link: <https://www.figma.com/community/file/917836295685422238>

# License:

Copyright © 2023 [Your Name or Company Name]. All rights reserved.

/\*

\* Copyright © 2023 [Your Name or Company Name]. All rights reserved.

\* This code is proprietary and may not be used or distributed without permission.

\*/  
  
Example:

Proprietary Software License Agreement

This Proprietary Software License Agreement (the "Agreement") is entered into by and between [Your Name or Company Name] ("Licensor") and the end user ("Licensee") of the software application (the "Software").

1. Grant of License

Subject to the terms and conditions of this Agreement, Licensor grants Licensee a non-exclusive, non-transferable, non-sublicensable, and revocable license to use the Software on a single device, solely for Licensee's personal, non-commercial purposes.

2. Ownership

Licensor retains all right, title, and interest in and to the Software, including all intellectual property rights therein. No title to or ownership of the Software, or any intellectual property rights associated therewith, is transferred to Licensee under this Agreement.

3. Restrictions

Licensee shall not:

a. Copy, distribute, or share the Software;

b. Modify, adapt, or create derivative works of the Software;

c. Reverse engineer, decompile, or disassemble the Software;

d. Remove, alter, or obscure any copyright, trademark, or other proprietary notices within the Software;

e. Use the Software for any commercial purpose or in any manner not expressly permitted by this Agreement.

4. Termination

This Agreement is effective until terminated. Licensor may terminate this Agreement immediately and without notice if Licensee breaches any of the terms and conditions herein. Upon termination, Licensee must cease all use of the Software and destroy all copies in its possession or control.

5. Disclaimer of Warranties

The Software is provided "as is" and without warranty of any kind. Licensor disclaims all warranties, express or implied, including but not limited to warranties of merchantability, fitness for a particular purpose, and non-infringement.

6. Limitation of Liability

In no event shall Licensor be liable for any damages, including but not limited to direct, indirect, incidental, special, consequential, or punitive damages, arising out of or related to Licensee's use of or inability to use the Software, even if Licensor has been advised of the possibility of such damages.

7. Governing Law

This Agreement shall be governed by and construed in accordance with the laws of [Your Jurisdiction], without regard to its conflict of laws principles. Any disputes arising out of or related to this Agreement shall be resolved exclusively in the courts located in [Your Jurisdiction].

8. Entire Agreement

This Agreement constitutes the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior and contemporaneous agreements, understandings, negotiations, and discussions, whether oral or written.

By using the Software, Licensee agrees to be bound by the terms and conditions of this Agreement.