

Geomapper

Students: Alexander Delgado
Mentor: Antonio Hernandez
Instructor/Faculty: Masoud Sadjadi, Florida International University

PROBLEM

Whenever I would code in the project I would have difficulty loading things, upon investigatin it was an operating system and dependcy error.Ensure the React Native application is compatible across various devices and operating systems, providing a consistent user experience.

Research

Starting the project came with its own challenges, and one of the most critical was ensuring cross-platform compatibility. Early in the development process, it became evident that limiting the application to a single operating system would alienate a significant portion of potential users. Focusing on compatibility was not just a technical necessity but a strategic decision to maximize the application's reach and user adoption across different platforms. React native can have issues running on Mac, so using Xcode can help, Xcode is a IDE for developers on mac devices. However, Xcode alone is not enough to help. To assist this research, the project needed to be understood beyond the code and breaking down how the project is affecting the system it is running on. The react native website goes over methods troubleshooting react native projects.

REQUIREMENTS

To ensure an application that works on both windows and mac computers. The solution needs to address current and futures issues that can arise when developing projects. The result also needs to not affect any of the existing APIs.

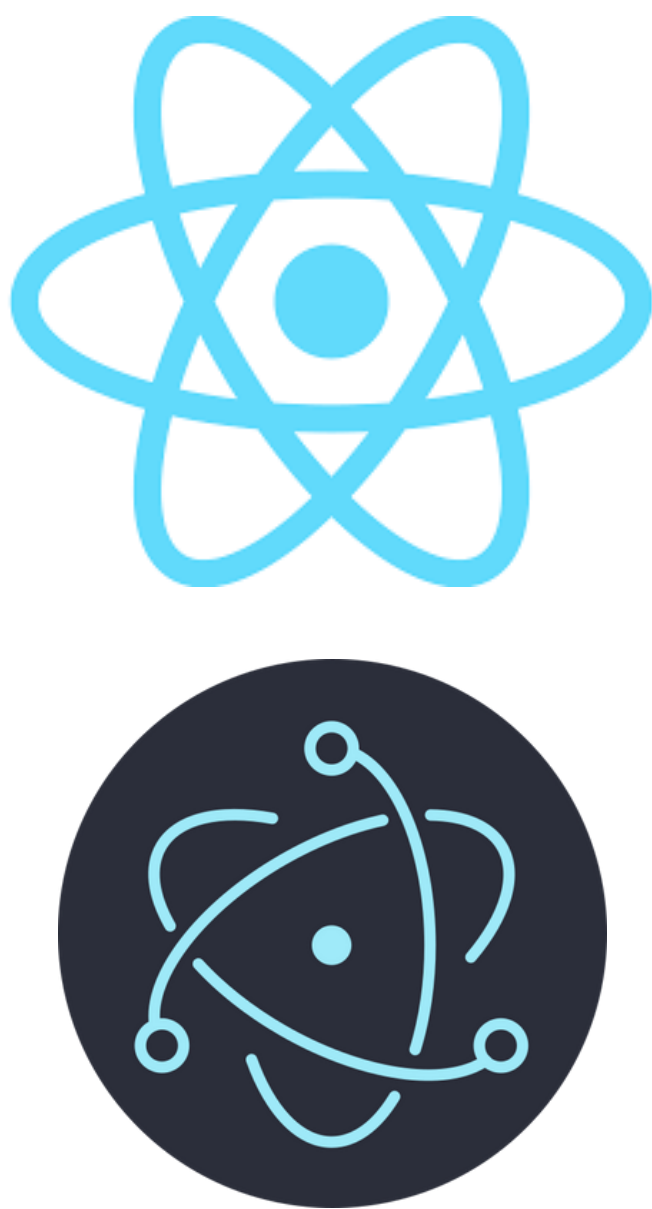
OBJECT DESIGN AND IMPLEMENTATION

Implementing measures for cross-platform compatibility helps users run the program better. These measures take us closer to a full packaged application. Using custom files like .xcworkspace help install dependencies specidic to Mac, then these can be checked in the package.json file. The package.json file is a file that gives the list of dependencies being used in the project, this can serve as a control when adding or deleting dependencies as the project moves forward. Writing scripts for users to manullay test their application also works, the scripts can either run the application or product helpful errors that can guide users for troubleshooting. Electron is an open-source framework that assists in cross platform compability for desktop applications. It does this by using tools that are already compatible with all devices. Electron is ideal for people with limited react native experience but still want to learn how to run their applications effectively.

SUMMARY

Ensuring cross-platform compatibility in a React Native project posed challenges, particularly with loading issues caused by operating system and dependency errors. Early on, it became clear that supporting multiple platforms was essential to avoid alienating users. Tools like Xcode for macOS, .xcworkspace files for dependencies, and the package.json file for managing dependencies helped stabilize the project. Custom testing scripts also improved troubleshooting. Additionally, Electron offers a simplified solution for developers with limited React Native experience, enabling seamless cross-platform desktop applications using familiar web technologies.

Technologies Used



REFERENCES

- 1. <https://reactnative.dev/architecture/overview>
- 2. <https://developer.apple.com/documentation/xcode>
- 3. <https://www.electronjs.org/blog/apple-silicon>

ACKNOWLEDGEMENT

I/We thank Antonio Hernandez for his/her/their assistance and mentorship that I/we received throughout the senior design project.