Assigment 2: Cross-platform Mobile Development

Annamária Váradyová
School of Computer Science, Social Media
and Web Technology, Linnaeus University, Sweden
av22xc@student.lnu.se

Implementation Details

1. Introduction of selected frameworks – React Native with Expo

1.1. Supported Mobile Platforms

React Native is a framework for building mobile applications using JavaScript and React. It enables programmers to create code once and deploy it across a variety of platforms, such as iOS, Android, and the web. React is a JavaScript library for creating user interfaces, and React Native is built on top of it. [Eisenman, B. (2015)]

A framework for creating mobile applications that work on both iOS and Android is called React Native with Expo. Expo is a collection of React Native-based tools and services that makes it easier to create and distribute mobile apps. It offers a set of resources that help programmers create, distribute, and maintain React Native apps. [Zammetti, F. (2018)]

Expo offers a variety of features and capabilities that facilitate the quicker and easier development of mobile apps, and it supports both the iOS and Android platforms. [Zammetti, F. (2018)]

1.2. Acces to device – specific features

Using a collection of APIs that are integrated into the React Native framework, React Native offers access to device-specific functionalities. These APIs give developers access to a variety of platform- and device-specific features and functions, including the camera, geolocation, and more. [Boduch, A., Derks, R. (2020)]

A wide range of device-specific functionalities are covered by React Native's extensive set of APIs, which includes: [Boduch, A., Derks, R. (2020)]

- *Camera: To access the device's camera and take pictures and films, developers can utilise the CameraRoll and ImagePicker APIs.*
- *Geolocation: With the help of the Geolocation API, programmers can access a device's geolocation and get the user's current location.*
- *Permissions: With the PermissionsAndroid and PermissionsIOS APIs, developers can ask for and manage permissions for various device capabilities, including the camera, microphone, and location.*

There are a variety of third-party libraries and modules that give access to extra device-specific features and capabilities in addition to these fundamental APIs. [Boduch, A., Derks, R. (2020)]

1.3. Development environment

You must perform the following actions in order to set up a development environment for React Native with Expo: [*What Is Expo?*, n.d.]

1. *Install Node.js: React Native with Expo requires Node.js to be installed on your PC.*
2. *Install Expo CLI by following these instructions: Expo CLI is a command-line tool that enables you to develop, launch, and manage React Native projects using Expo.*

3. *Code editor: To write and edit your React Native code, you will need a code editor. It was Visual Studio in our scenario.*
4. *Construct a fresh Expo project.*
5. *Launch the development server for Expo.*
6. *Start the application on a device or emulator: You can use the Expo Go app or the built-in iOS or Android emulators to run the app on a real device or an emulator. Download the Expo Go app from the App Store or Google Play Store, then scan the QR code produced by the Expo development server to launch the app on a physical device. Follow Expo's guidelines in the Expo Development Tools to launch the app on an emulator.*

The Expo APIs and services can be used to begin developing your React Native apps once your environment has been set up. [*What Is Expo?*, n.d.]

1.4. GUI Design

User experience, visual design, and platform-specific rules must all be carefully taken into account while designing a GUI for a React Native app. You may design a user-friendly and aesthetically pleasing GUI that engages and delights your users by adhering to these best practises and suggestions. [Boduch, A., Derks, R. (2020)]

- *Adopt a consistent design language: Decide on a design language for your app, such as Material Design or the iOS Human Interface Guidelines, and follow it consistently. Users will find it simpler to navigate your app as a result of the consistent look and feel that is created.*
- *Make it simple: Avoid overstuffing your Interface with functionality or pieces. Instead, concentrate on the main features and functions of your app and make sure that users can quickly reach them.*
- *Create your GUI responsively and with the ability to adjust to various screen sizes and orientations. Test your app on many platforms to make sure it runs smoothly and looks good on each one.*
- *Test your design: Do a usability and effectiveness test on your GUI using actual users to gather feedback. Make use of this criticism to enhance and improve your design.*

1.5. Ease of development and learning curve

Particularly for developers who are already experienced with web programming and JavaScript, React Native is renowned for its simplicity of use and relatively low learning curve. [Boduch, A., Derks, R. (2020)]

Developers, especially those who are new to developing mobile apps, can learn more quickly and efficiently by using React Native with Expo. [Panagia, S. (2022)]

These are some examples of how React Native with Expo makes development simpler and more available: [Panagia, S. (2022)]

- *Pre-made components and APIs: Expo offers a selection of UI components and APIs that are already constructed for functionality found in most mobile apps, like push notifications, camera access, and location tracking. As a result, developers don't have to spend time and energy creating these functionality from scratch.*
- *Easier setup: Expo offers a quick, straightforward setup procedure that does not require developers to set up native build tools like Xcode or Android Studio. For*

developers who are new to creating mobile apps and may not be familiar with these tools, this can be extremely beneficial.

- *Testing on actual hardware: With the Expo Client app, Expo offers a simple method for testing your programme on actual hardware, negating the need to set up emulators or actual hardware for testing.*

React Native is usually thought to be simpler to learn and develop with than some other native app development frameworks, even though there is still a learning curve to mastering it. React Native with Expo is an excellent option for developers who are new to the industry or want to quickly prototype and produce native apps without a steep learning curve because it may drastically minimise the complexity and learning curve of mobile app development. [Panagia, S. (2022)]

1.6. Distribution

For developers, React Native with Expo offers a few extra distribution options that can streamline the deployment procedure: [Panagia, S. (2022)]

- *Over-the-air (OTA) upgrades for the Expo Developers can deploy updates to their apps via Expo's over-the-air update service without going through the formal app store review procedure. This can be helpful for swiftly correcting bugs or introducing new features. Users will automatically receive the latest version the next time they access the app thanks to Expo OTA updates, which are initiated once a new build is published to Expo.*
- *Expo App Store builds: With the help of Expo's build service, developers may produce iOS and Android app builds without having to set up Xcode or Android Studio. Via Expo's distribution service, Expo App Store builds may be sent to the Apple App Store or Google Play Store, simplifying the app store submission procedure.*
- *Over-the-air installation: Expo further enables developers to distribute their programme to consumers via a web link that enables over-the-air installation of the software without the need for an app store. This can be useful for internal distribution within a company or for providing beta copies of an app to testers.*
- *Web deployment: React Native with Expo offers web deployment, which can be viewed on desktop or mobile devices via a web browser. For developers who wish to construct a mobile app that also functions on the web or for prototyping and testing the app's functionality, this can be a beneficial choice.*

React Native with Expo, in general, offers a variety of choices for app distribution that can streamline the deployment procedure for developers. Expo OTA updates, App Store builds, over-the-air setup, and web deployment all aid in streamlining the distribution of apps, enabling developers to distribute their products to users more quickly and effectively. [Panagia, S. (2022)]

Implementation of the mobile application

With React Native and Expo, we created a mobile app.

We establish a new project with the required configuration files and dependencies using the Expo CLI.

Dependencies were added:

- *Expo-location[1] module enables developers to access the device's location information in React Native apps. It offers a simple-to-use API for getting the device's current position, tracking location changes, and keeping track of the user's movements.*
- *Expo-camera[2] is a potent tool that enables programmers to create camera-based programmes that offer rich and interesting user experiences. Expo-camera is a crucial module for camera-based React Native applications since it makes it simple for developers to access the device's camera without having to deal with challenging camera APIs.*
- *Expo-media-library[3] module, which enables developers to access the device's media library in React Native apps. Developers may quickly retrieve and manage media assets like images and movies saved on the device by utilising Expo-media-library to access the media library on the device.*
- *Expo-sharing[4] offers a variety of sharing techniques, including sharing via the system user interface, sharing with other apps, and sharing with contacts. Additionally, it offers choices for personalising the sharing process, such as the ability to add a message or subject to the shared item and designate the kind of content that should be shared.*

Images and locations are stored in Firebase[5] for the purpose of storing our user-generated data.

Google created the mobile and web application development platform Firebase. It provides a range of resources to assist programmers in creating dependable, scalable, and high-quality programmes.

Firebase is a backend-as-a-service (BaaS) solution that takes care of an application's infrastructure and server-side functionality so that developers can concentrate on front-end development. Real-time databases, authentication, hosting, storage, cloud services, and machine learning are just a few of the features offered by Firebase.

Source code: https://github.com/Anna8295/mobileApp
Source video: https://youtu.be/CslCwL8t6UY

Discussion

Creating a mobile application was a new experience for me. Since I only have basic knowledge of programming, I wanted to continue in a similar programming language. For this reason, I chose React Native, but after studying and knowing how to implement a project, I combined React Native with Expo. The entire program is written in the React language, but the environment is much easier to implement. The installation itself and at the same time the test using the application on the mobile phone.

References

Eisenman, B. (2015). *Learning react native: Building native mobile apps with JavaScript*. " O'Reilly Media, Inc.".

Boduch, A., & Derks, R. (2020). *React and React Native: A complete hands-on guide to modern web and mobile development with React. js*. Packt Publishing Ltd.

Zammetti, F. (2018). *Practical React Native: Build Two Full Projects and One Full Game Using React Native*. Apress.

[1] https://docs.expo.dev/versions/latest/sdk/location/
[2] https://docs.expo.dev/versions/latest/sdk/camera/
[3] https://docs.expo.dev/versions/latest/sdk/media-library/
[4] https://docs.expo.dev/versions/latest/sdk/sharing/
[5] https://firebase.google.com

What is Expo? (n.d) Expo documentation. Retrieved February 27, 2023, from https://docs.expo.dev/introduction/expo/

Panagia, S. (2022). *What is Expo and why it matters for app development — Moze*. Moze. https://www.mozestudio.com/journal/what-is-expo-and-why-it-matters-for-app-development/