Bible in Living Sound

A Mobile Application

Quillan Jacobson and Charles Lambert

WWU Department of Computer Science

Walla Walla University
Seventh-day Adventist Higher Education

ABSTRACT

The purpose of this project was to develop a cross-platform mobile application for the Bible in Living Sound. The app is built in React Native. It plays audiobooks from cloud and local storage, and has user authentication. This is accomplished using several libraries such as Redux (a data store), React Native Fetch Blob (file fetch) and React Native Video (audio playback).

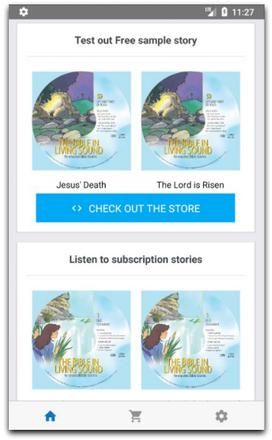
INTRODUCTION

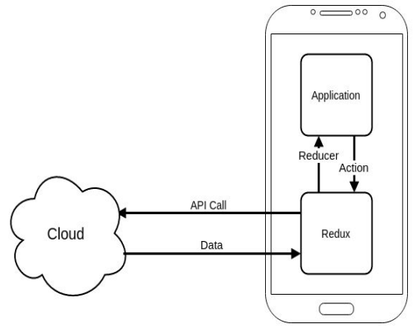
The Bible in Living Sound is a Christian audiobook company. They provide dramatized versions of Bible stories in several languages internationally. Their current format for distributing these stories is CD. In order to move away from this format they asked us to create a mobile application. The application was developed for iOS and Android React Native was the natural choice for this project, as it allows Javascript in the form of JSX to be compiled down to the native languages of each respective platform. This app was developed to be a streaming and downloading service for audio Bible stories. It provides user creation and authentication, sudo purchases, is designed to meet a professional standard, and drives new visitors to the Bible in Living Sound's website.

DATA FLOW AND UX ANALYSIS

The application consists of four main pages and three conditional pages. The home page presents subscriptions and general information, the store and owned pages manage audio files and audio playback, and finally the settings page handles login and payments.

The highest level components are the navigator and the redux store, which handle navigation and the persistence of audio playback across the application respectively. Audio is handled at the same level to avoid duplicating the audio playback component.





The data flow for storage is done using Redux and the application's local file system. Redux allows the application to dispatch actions that will either make API calls or produce data, which is then passed to reducers that in turn update the application state. Redux in the form of Redux-persist is used to store some of the application state between sessions, such as the authentication token and the lists for store, cloud, and local items.

CONCLUSION

React Native is a useful framework, but by no means is it simple or intuitive. It halves the workload of app development for cross platform applications and requires only slightly more effort to expand to a web page. Do not assume that React Native is like HTML and JavaScript. React Native is entirely JavaScript and has proven to have various edge cases and difficulties in development.

Future Work

Although several features were completed from the specification the overall app still requires real in application purchases, a backend service/API to provide audio files, subscription provider/service, and to register/publish the application to the Google Play store and the Apple App Store. In terms of graphic design, the application could also benefit from individual graphics for each story.

REFERENCES

- http://www.bibleinlivingsound.org/
- https://facebook.github.io/react-native/
- https://redux.js.org/
- https://github.com/react-native-community/react-native-video