Interactive Education Using Near Field Communication

by

Augustina C. Costanza

February 1st, 2026

A project write-up submitted to the
faculty of the Graduate School of
the University at Buffalo, The State University of New York
in partial fulfillment of the requirements for the
degree of

Master of Science
Department of Computer Science and Engineering

To my family, friends, and professors – for supporting me through the arduous process of getting up in the morning.

Abstract

The ever-changing world necessitates that education adapts with it. It would be simple to return to pen and paper, printed ink in a textbook. These tried-and-true methods have sustained humanity for centuries. As progress marches ever forward, however, it becomes futile to turn the clock back. People of today interact with the world differently with much blame to be placed on the shoulders of computer science. Rather than stand against the tides of change, it would be better to take full advantage of innovation and present learning in a way that's geared to present-day society.

That is, in a way, the goal of this project. The client approached the University at Buffalo with a request for an app that would provide an interactive scavenger hunt experience at the Niagara Aerospace Museum. This app is the product of that request. It uses near field communication (NFC) and React Native to provide a seamless experience across devices without having to worry about hardware or browser differences that would impede uniformity. React Native was determined to be the best way to achieve this as it has libraries for reading NFC tokens whereas iOS doesn't allow something as simple as for tokens to be scanned without reading the content as a URL. The app uses a Node.js backend, WebSockets, and a MySQL database.

TABLE OF CONTENTS

List of Figures

Introduction

This project was initially pitched to the University at Buffalo for CSE 611: Master's Project Development, where it had been undertaken by at least two teams before it was adopted for a master's capstone project. The goal of the project was to monopolize on the trend of museums shifting towards interactivity in unprecedented ways: smartphones (Ceipidor et al., 2013). The use of near field communication would be able to give patrons a level of engagement with the museum's exhibits and auxiliary materials that simply wasn't achievable before. In particular, the app that would be developed would be a scavenger hunt app that would scan NFC tokens on exhibits to check off items on a list of questions. Patrons would be able to play either on their own or in a group, like how games are created and played with Jackbox and Kahoot. The app would also be able to provide information on the exhibits such as trivia and patrons could request hints if they were struggling with any questions. All screens of the app have been provided in the index for reference.

An earlier version of the app was made for use through a mobile browser and used raw HTML, Firebase, and a simple database from the documentation that was able to be recovered. The version prior to its transition into a capstone project was also a web application and used a React frontend and Node.js backend with WebSockets for real-time communication. While this made development easier in terms of styling and code reusability, the project ultimately fell short due to browser differences and hardware limitations. These issues were rooted in a key part of the app: near field communication, or NFC.

Near field communication, in a sense, can be taken as QR and RFID's younger, more specialized sibling. Both had their merits but were susceptible to limitations that necessitated the creation of something more cost-friendly and easier to adopt. For QR codes, things such as lighting would made it impossible for them to be properly used. For RFID, the cost wasn't worth it in many instances, averaging between 10 cents and $1, and mobile devices didn't have the

capability to read RFID at the time NFC was being considered (Want, 2011, Technology Solutions). Existing as a subset of RFID, NFC operates at high frequencies, supporting "communication between an active reader and a passive tag" and taking advantage of its modern-day integration in smartphones (Want, 2011, Features of the NFC Standard).

If NFC has now become the apple of seamless information sharing's eye, why was it presenting a roadblock for the development of this app?

The answer is simple: browser differences and hardware limitations. While most Android devices can read NFC tokens without hassle and have Google Chrome as their default browser, issues arose when catering to iOS. Whereas Android devices can read the content of any tags, iOS devices can only read URLs from tags and Safari (Apple's native browser) simply would exhibit different behavior than Chrome – opening extraneous tabs, ignoring broadcasted events, etc. Apple devices also need to have NFC explicitly allowed by some app to do more than read URLs from NFC tags. This simply wasn't capable within a browser. Since the version of the app before the one presented in this paper was solely a web application intended for use only on smartphones, this required tab management that wasn't intuitive as most browsers don't even allow JavaScript to close open tabs and the game would continue running elsewhere without the knowledge of the user. It was an unwieldy, untamable beast.

Thus, the solution to this issue was to rebuild the web app's frontend entirely, transform it into a React Native app, and build it with Expo.

Problem Solution

The following section will give a brief overview of the main components of the project and discuss their integration in depth.

React Native

React Native is a version of React that is geared toward multi-platform development. While React is strictly a web application framework, React Native can be utilized to create native

apps for Android, iOS, and the web. The bulk of the time on this project was spent taking code written in React and translating it into React Native. Clearly, this wasn't an impossible task, but there were many browser staples that had been taken for granted that were unusable in a mobile application. Browsers use cookies and the like to track client-side data and match users with their data in persistent storage on the server-side. This is very important when trying to manage games between multiple mobile devices that navigate through a given app's subpages. Additionally, changes needed to be made to the structure of the app to allow for the integration of NFC, as well as some accommodations since the libraries used were third-party to Expo.

Expo

Expo is a React Native framework that allows for developers to either run their applications within a sandbox using their app (Expo Go) or upload their code to Expo's servers to create development or production builds. These builds make it possible to download versions of the app onto test devices or into simulators. Typically, a React Native app would only really need to be tested in Expo Go, but due to the library used for NFC reading not being part of Expo and instead being a React Native module, the sandbox version of the app could not be utilized for testing (Revtel, 2023). Instead, testing was done entirely with a development build on an Android with NFC capabilities and a matching version of the app. As Android is open source, building and testing the app on this device was free compared to iOS, which would've required joining the Apple Developer Program for a hefty fee. Put formally, this was not conducive to prolonged financial prosperity when developing an app once again in its infancy.

MySQL

Once the difficulties with converting the app and NFC were sorted out, it was a matter of testing. To save on AWS costs, a local MySQL database was used throughout development, though it would be simple to transfer everything over to proper online hosting. As it's impossible

to connect to a local database on a host machine while running a React Native app through Expo using typical means, the host machine's IP address was utilized as an endpoint in testing.

NFC

Previously, the web application versions of the app used NFC tokens that contained URLs. As it had been recreated as a mobile app instead, NFC could be enabled within the app, like how Apple Pay or Google Pay transactions take place in a checkout line, which allowed different data to be written to the NFC tokens. The nuisance of URL parsing was simplified into reading a text record that contained a singular integer representing the exhibit in the corresponding table in the database.

Conclusions

At the time of this write-up, the app has a complete Android build that, following further testing and re-integration with AWS, will be published on the Google Play store. The next step, once it's taken on by another team or individual for CSE 611 or independent study, will be to create and test the iOS build. Following that, installation can begin at the Niagara Aerospace Museum in Niagara Falls, New York. Testing in-person with exhibits that can be seen is integral to the development of the app and gearing it to the user experience.

This project initially was being worked on in tandem with an inventory management system for the Niagara Aerospace Museum and utilizes the same database for many of its operations. It would be remiss to not continue to take advantage of that and provide auxiliary materials separate from the scavenger hunt, though this is merely a suggestion. Whoever takes this on will hopefully have an amount of passion for frontend and game development.

Works Cited

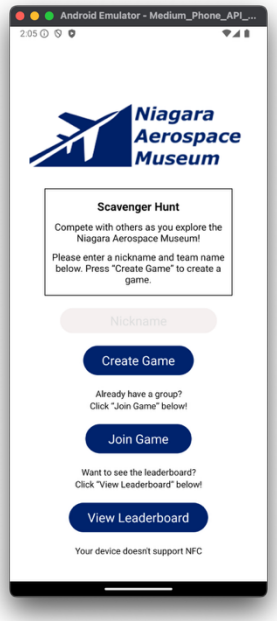
Ceipidor, U. B., Medaglia, C. M., Volpi, V., Moroni, A., Sposato, S., Carboni, M., & Caridi, A. (2013). NFC technology applied to touristic-cultural field: A case study on an Italian museum. *2013 5th International Workshop on Near Field Communication (NFC)*, 1–6. doi:10.1109/nfc.2013.6482445

Revtel. (2023). Expo Go. Retrieved from https://github.com/revtel/react-native-nfc-manager/wiki/Expo-Go

Want, R. (2011). Near Field Communication. *IEEE Pervasive Computing*, *10*(3), 4–7. doi:10.1109/mprv.2011.55

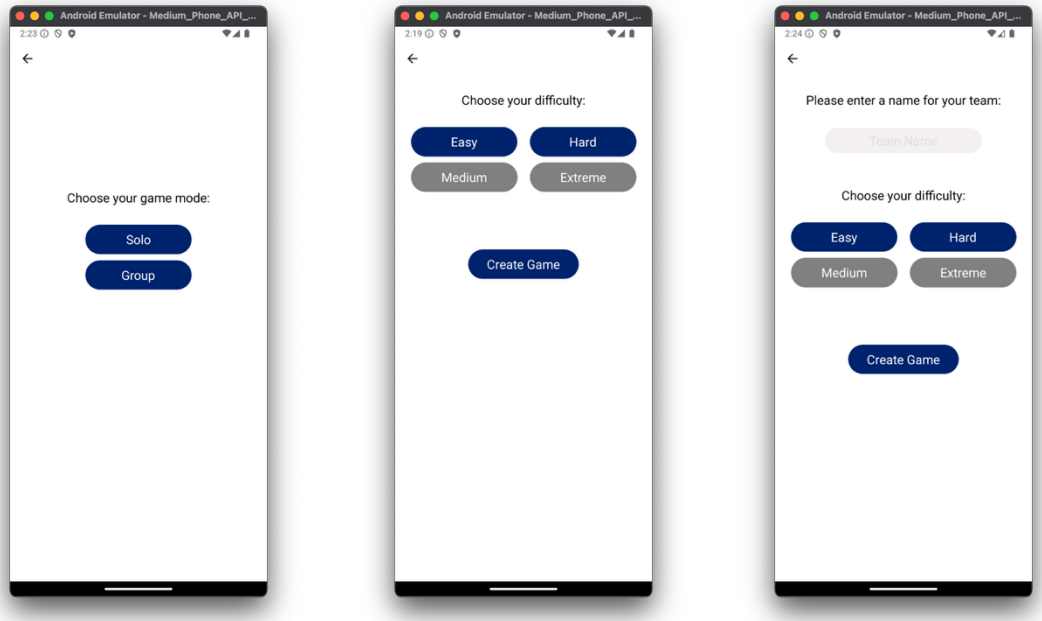
Appendix

Figure A1. App Start Screen



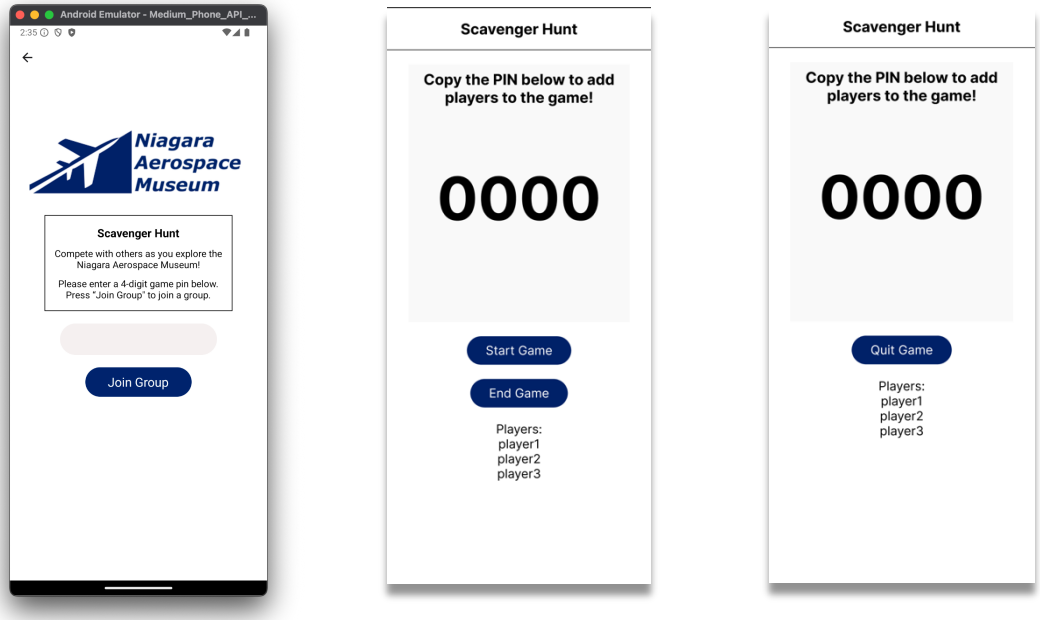
Note. The Niagara Aerospace Museum logo belongs to the Niagara Aerospace Museum.

Figure A2. Game Settings



Note. The screen for choosing a game mode (left), creating a solo game (middle), and creating a team (right).

Figure A3. Joining and Starting a Game



Note. Joining a game with a PIN (left), the screen the game creator sees (middle), and the screen a regular player sees (right).

Figure A4. View Leaderboard

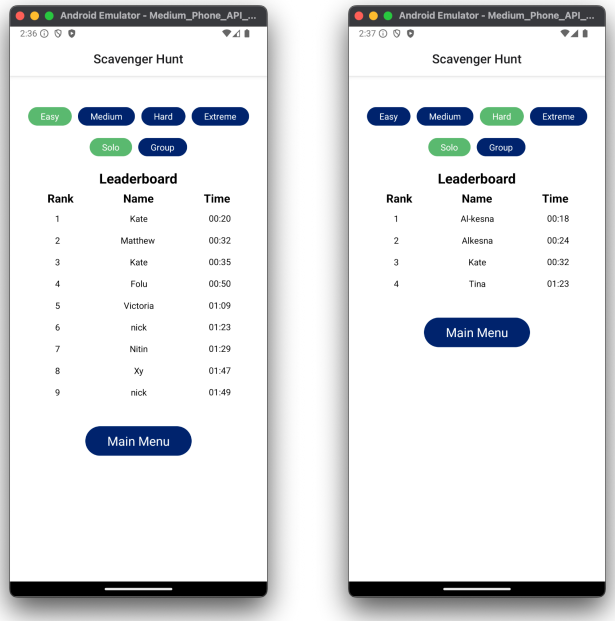


Figure A5. Scavenger Hunt List

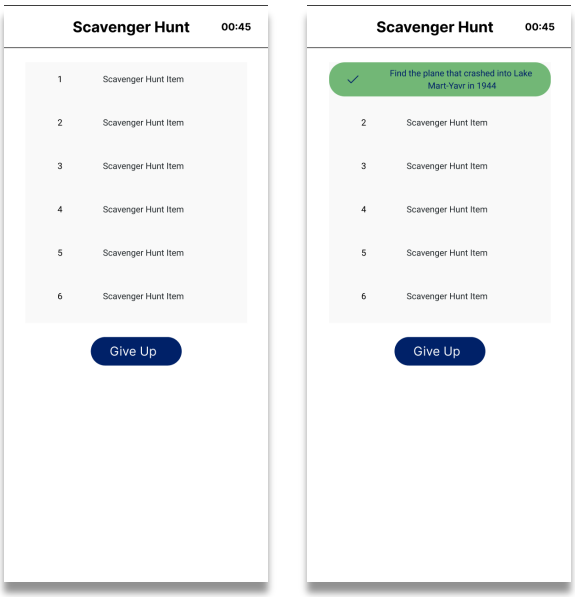
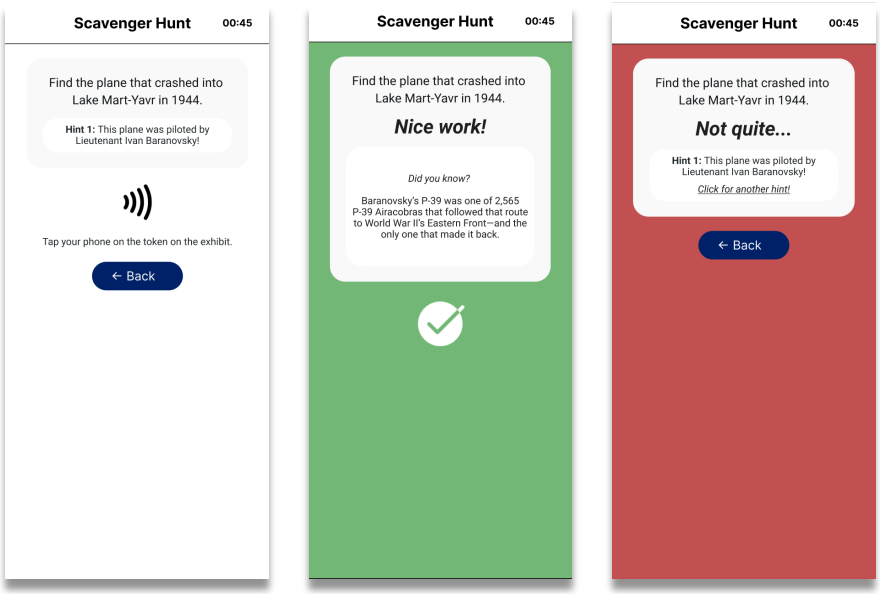
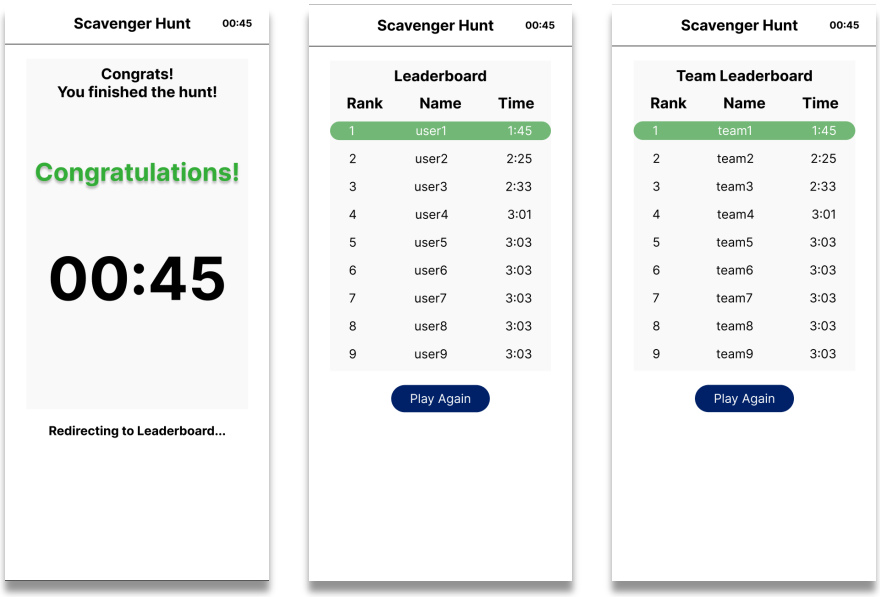


Figure A6. Answering a question.



Note. After choosing an item on the list (left), answering correctly (middle), answering incorrectly (right).

Figure A7. Completing the Game



Note. The screen shown after a game is complete (left), the leaderboard for solo games (middle), and the leaderboard for team games (right).