



Subodh Dahal

Building User Application for Drink Vending Machine using React Native

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

1 March 2021

Abstract

Author: Subodh Dahal
Title: Building User Application for Drink Vending Machine using React Native
Number of Pages: 27 pages
Date: 1 March 2021

Degree: Bachelor of Engineering
Degree Programme: Information Technology
Professional Major: Software Engineering
Supervisors: Janne Salonen, Head of Department ICT

The thesis was centred around the unorthodox practice of developing a client-side user-application for drink vending machine fitted with Android OS using React Native. The main requirement included the identification of the communication practice and protocols between the CPUs in the machine, an application to register and identify valid customers and their subscription status with the help of RFID tags attached to their personal protein shakers and serve the desired drink and report every transaction to the server. The invoicing would then be taken care of differently.

The project was done for OptiShake Oy, which envisioned the use of protein and recovery drink vending machines on the gyms where the gym goers can get the desired protein and recovery drinks. The machines were especially customized and manufactured in China and shipped to Finland. The default application on the machines required an active connection to the servers of the manufacturer thus the need for independent application had arisen.

Development was carried out using React Native with use of multiple Java based modules. While the perceived difficult parts of controlling the hardware using JavaScript was already done, the company decided to change the requirements and switch to Java to develop the application. Thus, the project was cut short. The thesis reports on the challenges faces and overcame during the development process and the findings.

Keywords: React Native, Drink Vending Machine, User Application

Contents

1	Introduction	6
1.1	Business Context	6
1.2	Business Challenge and Objective	6
1.3	Project Requirements	6
1.4	Research Questions	7
1.5	Thesis Structure	7
2	Background	8
2.1	Vending Machines	8
2.2	Coffee Vending Machine used by case company	8
2.3	Android Application	10
2.4	React and React Native	10
2.4.1	React	10
2.4.2	React Native	11
3	Project Requirements	13
3.1	Research	13
3.2	User App to be Used in the Android Machine	13
3.2.1	Working with the hardware	13
3.2.2	Configuring Drinks	14
3.2.3	User Registration	14
3.2.4	Reporting	14
3.2.5	Error Handling	15
3.2.6	Displaying Information	15
3.2.7	User Status Checking	15
3.2.8	UI Translatability	16
3.2.9	Offline Operations	16
3.2.10	Admin Operations	16
3.2.11	Fault Reports	16
4	Development Approach, Tools, and Implementation	17
4.1	Approach	17

4.1.1	Pre-Study	17
4.1.2	Collecting User Stories	17
4.1.3	Schema Design	18
4.2	Tools	18
4.2.1	Visual Studio Code	19
4.2.2	Android Studio	19
4.2.3	Git and GitHub	19
4.2.4	Trello	20
4.2.5	ADB	20
4.3	Implementation	21
4.3.1	React Native	21
4.3.2	Redux and React-Redux	21
4.3.3	Android Native Modules	22
4.3.4	Other NPM packages	22
5	Outcome	22
5.1	User Application	23
5.1.1	Developed Features	23
5.2	Difficulties	24
5.3	Future of the Project	25
6	Conclusion	25
	References	26

List of Abbreviations

UART:	Universal asynchronous receiver-transmitter
RFID:	Radio Frequency Identification
ADB:	Android Debug Bridge
AVD:	Android Virtual Device
NPM:	Node Package Manager
JWT:	JSON Web Token
UI:	User Interface
MVP:	Minimum Viable Product
OS:	Operating System
SDK:	Software Development Kit
JDK:	Java Development Kit
DOM:	Document Object Model
SCM:	Source Code Management

1 Introduction

The thesis explores the process of building a user application for the protein drink vending machine used by OptiShake Oy in several gyms. The case company was started in 2019 with the goal of providing gym goers the ease of having different selection of protein drink at their disposal without having to carry it with them.

1.1 Business Context

The idea was to equip the gym with the drink vending machines which had several protein powders from which the customers can make their protein shake in the gym. The goal of the company was to serve unlimited amount of drink on monthly subscription basis. The objective of the project was to explore the possibilities to develop a full-fledged system for the company using Javascript, by developing an MVP for user application.

1.2 Business Challenge and Objective

The case company is currently operating on three different gym on a slightly different model vending machines but with arrival of new customized vending machines from China, the company is looking to modify and expand the operations. The current application in the new vending machines fail to function properly without constant connection to the server of the manufacturer which is a major problem for the company. The project aims to solve this problem with addition of other features like customer authentication using RFID and reporting of the data and transactions in desired manner with the new application.

1.3 Project Requirements

In addition to the aforementioned solution, the project required other tailored features. RFID identification to authenticate the paying customers and their

subscription status was one very important feature for the MVP. The reporting of every transaction where it can be easily monitored and used to invoice the customers was another priority. Other features like displaying offers, advertisements and making fault reports were desired for the application although deemed not essential for MVP.

The thesis aims to clarify the project by explaining the background of the project, the specific technical requirements, the approach with which the problem was tackled, the methodology and the technologies used throughout the process to get the final outcome and the conclusion.

1.4 Research Questions

As the project is of an experimental nature and the carried out with relatively new technology to solve a unique problem, the motivation was to find out how React Native would perform in such scenarios. Two main questions for the research was:

1. How would React Native perform when it comes to interacting and controlling hardware?
2. Does using React Native make the overall development process easier for such problems?

As React Native is very new technology and especially not used for application like this, the research aimed to explore the possibilities to implement the technology to solve similar problems in the future.

1.5 Thesis Structure

This thesis is structured to guide readers from the background of the field, technology and company to the requirements of the project, development approach and implementation to the result and conclusions. Every section gives some relevant theoretical knowledge and its use in the project.

2 Background

2.1 Vending Machines

The vending machines have been in use since the Greek mathematician Hero invented one to dispense holy water inside Egyptian temples. Vending machines have existed in a form or another to serve many different products like newspapers, cigarettes, drinks, fruits, foods etc throughout the world. Drink vending machines can also be found easily in airports, bus stops and kiosks to dispense hot and cold drinks as required by the customer. With the introduction of credit card scanners and contactless payments, the possibilities and ease of vending machine use have increased drastically. It is one of the most prominent ways to sell consumer products in present time[1].

2.2 Coffee Vending Machine used by case company

Coffee vending machine, invented by Rudd-Melikian company of Philadelphia, came to market in 1946 in Kwik Café which dispensed hot coffee by mixing instant coffee and hot water into a paper cup[2]. The machine used by case company to serve protein drink follows the same principle. The machine serves protein powder and water into the shaker according to the customers' needs.



Image 1: Front Part of the machine showing the taps and cannister space

As shown in figure 1, the machine contains water pumps with five outlets and space for five separate protein powder cannisters which would dispense the powder and water separately into the shaker with right command.



Image 2: Back Part of the Machine

The machine has two processors. The one in the back, as shown by fig 2 above, controls the hardware of the machine including boiler, water pumps and cannister motors. The front part of the machine is an android tablet with Rockchip RK3288 with ARMv7 processor running Android 8 as OS. The user application built as a project would be run on this device.

2.3 Android Application

Android is an Operating System initially developed by a silicon-valley start-up Android Inc, which was acquired by Google in July 2005. The Google has since been releasing different versions Android which is widely used in devices of different brands, shapes and sizes. Android is an open source platform, meaning it is not tied to one brand of devices and the source code is open to view and modify for everything which allows Android to be flexible and modifiable OS that it is today. Android runs over 70% of all the mobile devices worldwide which makes it an attractive platform to develop and operate on[3][5].

Android application is an application that allows user to perform certain tasks with the device. Usually Android application is developed using an IDE like Android Studio or Eclipse and Android SDK which uses Java Android Library. Different Java libraries are used in the development process according to the task required[4].

2.4 React and React Native

React was the main technology used for the project. Some theoretical information and background of React and React Native are explained briefly in this section.

2.4.1 React

React is an open source JavaScript library released by Facebook in 2013 with the purpose of building UI easier for frontend application with ever growing complexities and states. React revolutionized front-end development with three key concepts: Declarative, Component-based, Stack independent. With react, a separate view is designed and rendered for each state which in unison form a user interface. Each component is coded separately with control of its state which can receive the data to be rendered when state is updated. It is possible with use of JSX instead of HTML template. React is compatible with wide range of technology stack for the development of user interface.

React implements the concept of Virtual DOM. DOM is a representation of UI in a tree structure; traditionally a page was rendered and after the change the tree was updated and re-rendered. Virtual DOM renders the part of the UI or node of the tree once it is loaded and is updated dynamically to accommodate the change in the state of the DOM[6][8].

Most basic React Component looks as follows:

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Image 3: A simple example of React code as presented in official guide [7]

2.4.2 React Native

Facebook introduced React Native in 2015 to develop their mobile application after success of React on developing the web application. The main goal of React Native was to eliminate the need to learning multiple language to create mobile applications for iOS and Android. React Native allows rendering UI in mobile applications with same principle as React for web applications but unlike React, React Native does not use the WebView. React Native uses React Native bridge to asynchronously call the native widgets on each operating system. It invokes Views on Android and UI Views for iOS which allows the application to have same feel and functionality with same JavaScript code even in different platforms. This makes React Native better than other conventional JS framework for mobile applications like Ionic which uses WebView and lack the same level access to mobile's native UI components and APIs. React Native also allows the developer to see the change during development without having to compile the native application every time[3][9][10][17].

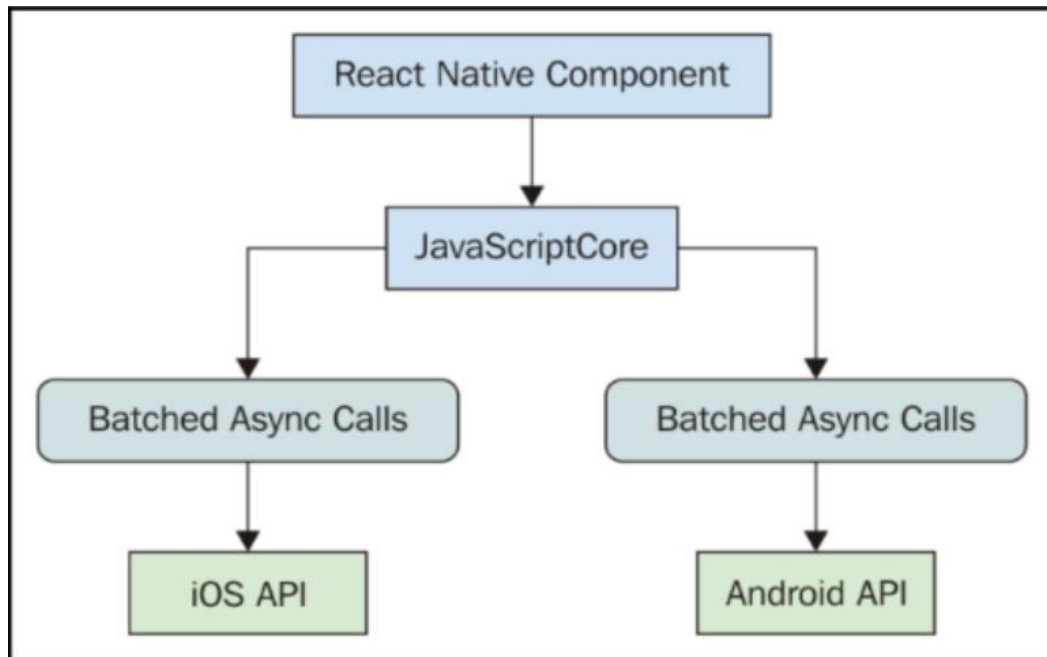


Image 4: The inner working of React Native[10]

Although made to be platform independent, React Native does have some platform specific components and codes. This is due to difference in UI components, widgets and their purpose in iOS and Android. Additionally, unlike in React, in React Native views cannot be rendered with web components like `<div>` or `` but native components from React Native libraries have to be used like `<View>` or `<Text>` as shown in the figure 5 below[9][10].

```

import React from 'react';
import { Text, View } from 'react-native';

const YourApp = () => {
  return (
    <View style={{ flex: 1, justifyContent: "center", alignItems:
"center" }}>
      <Text>
        Try editing me! 🐛
      </Text>
    </View>
  );
}

export default YourApp;
  
```

Image 5: Example code of React Native as presented in official guide

3 Project Requirements

3.1 Research

Due to the novelty of the project and absence of proper manual from the manufacturer about the machine, the project started with basic research about vending machine and the communication protocols used in them. The project required the understanding of the communication between the Android tablet in the machine with the processor which controls the pumps and motors in the back of the machine. The machine used UART port connection between two processor in it and communicated using hexadecimal codes. The basic codes structures and commands were provided in excel sheets. The information regarding operating the machine using JavaScript application, the ease, and the challenges to maintain and scale needed to be answered.

3.2 User App to be Used in the Android Machine

The desired final user application for the case company was a sophisticated and detailed application with different functionalities but regarding the project of this thesis, some requirements were deemed as the most crucial and must have in the MVP of the application. Requirements were discussed and were considered necessary for the MVP or in the future versions of the application. It was acknowledged that necessity for modification or addition in features would be apparent while on operation. Some requirements are listed below.

3.2.1 Working with the hardware

Due to the use of JavaScript as the main language for development, this was the most challenging feature to achieve and the most important one too. The application needed to be able to communicate with the back part of the machine to control the pumps and motors to dispense the right drinks properly. In addition

to controlling the hardware within the machine, the application also needed to work with an external RFID reader to scan the RFID tags in the protein shaker to identify and authorize customers.

3.2.2 Configuring Drinks

The application and the MVP required the ability to have the drinks configured from the server. In other words, when the application was started it would fetch the drinks, info, nutritional value, price, container number and recipe code from the server as JSON object and display the needed information on the screen.

3.2.3 User Registration

The feature to register new users was also very important and needed in the MVP. The registration would be done at the time of protein shaker purchase when the user would scan the shaker on the RFID reader attached to the machine and the valid shaker would be prompted to register for a user if not already registered to a certain user. In such case, user can buy drink, subscribe/unsubscribe or edit his details from the application.

3.2.4 Reporting

The application required to be able to record every transaction in the machine including user registration, modification of data, drink served, new subscription etc in organized manner and sent to server for processing. This was one of the main functions required as the business relies on the usage of machine and being able to process the service and invoice the customers accordingly.

3.2.5 Error Handling

Because the machines would be in the gym and not in a controlled office environment in presence of personnel, the machine could run into any errors or there could be other interruptions like jammed cannister, lack of running water supply etc, which meant the application needed the feature to listen to the response codes sent by machine and disable some or all drinks if necessary. This feature was considered necessary but not included in the MVP.

3.2.6 Displaying Information

Since the application was to be used in gyms and covered variety of functionalities, it was desired that the application would be able to behave differently and pass different information when needed. For example, in error situation, the application would display the error and the possible date for it to be fixed and what to do instructions for the customers, customized information about the drinks and possibility of introduction of new drink, gym specific greetings and news etc. If explored fully, the possibility of using idle screen time for advertisements and more were also discussed as way of boosting the company revenue.

3.2.7 User Status Checking

One desired feature for later versions was the ability for the application to check the payment status for a user and act according to the status. The application would allow users to use the service until certain period after the payment is delayed, remind them to pay the subscription fees and stop the user when the payment time limit is passed.

3.2.8 UI Translatability

The UI of the machines were to have at least two different language Finnish and English. This was for later revisions but was considered one the first things to be accomplished once in production.

3.2.9 Offline Operations

Because the application worked in a real-world environment without supervision, the chance of disruption in connection was considered as well. And because the application was to rely heavily on the information from the server to serve the customers, it would be a big problem if the connection was lost. Thus, it was discussed that application would use loaded and stored information from the application to serve the customers when the connection is lost and store the info locally until an API call can be made to the server to pass the new information.

3.2.10 Admin Operations

In subsequent releases of the app, it was deemed necessary to have admin area in the application from where operations could be run which are not allowed to normal users such as run basic pipes cleaning commands from the app, overviewing the sales and other transaction in the machine etc. The plan was to register some RFID tags for admins which would allow them to access the administrative area.

3.2.11 Fault Reports

Another desired features in the app was fault report system. For example, if a user ordered a drink and did not get drink due to any reason or if the amount of protein powder was lower then she should be able to report the fault. The fault report system was also not included in the plan for MVP.

4 Development Approach, Tools, and Implementation

This section illuminates the method by which the project was carried out. The first part Approach explains how the project was approached, Tools lists the tools and technologies used to implement the planned solutions and Implementation part outlines some application of the ideas.

4.1 Approach

This section explains the approach taken for the project.

4.1.1 Pre-Study

Due to the exact nature of the project being something not done or studied before, there was considerable amount of pre-study required to begin to tackle the problem. The basic research started with Google searches about the vending machine and the inner working and software of the vending machines in general. The case company had some documents containing info about the communication protocol in the machine and the required hexadecimal code generator to operate the hardware of the machine. Thus, it made the start somewhat easy as the start point for the research. The majority of this part consisted of reading blogs, guides, and watching YouTube videos and asking teachers and bloggers about the project.

Once the basic idea of the project was gained, the main plan was to run the base React Native app in the machine and run the basic commands through it to test if the machine acts as desired and to develop an actual application around that foundation.

4.1.2 Collecting User Stories

As the app was being developed to serve the customers, first task after basic research was to come up with the user stories:

- Users should be able to register using new shaker

- Existing users should be able to purchase new shaker
- Users should be able to order the drink of their choice
- Users should be able to modify their personal details
- Users should be able to subscribe or use single drink pay

The scenarios were discussed and agreed upon with the company staff beforehand.

4.1.3 Schema Design

The schema refers to the representational plan after which the JSON objects would be modelled in the code. As the information from and to the server would be passed in this form, it was very important to be detailed while making schema. The schemas were designed for user, drink, shaker and transaction for MongoDB.

Example of Drink object is shown the figure below.

```

2      {
3      "id": "Chocolate Drink",
4      "name": "Chocolate Drink",
5      "description": "Best Recovery Drink after workout",
6      "price": "1 Euro",
7      "machineCodes": {
8          "order": "AA550401110116",
9          "recipe": "AA551F151100C8015E0000000000000000000000000000000000000000000000006B",
10     },
11     "calories": 1589,
12     "unitsSold": 21553,
13     "ratingAverage": 4.7,
14     "reviewCount": 1294,
15     "drinkImgUrl": "https://cdn.verk.net/images/5/2_364772-1222x1593.jpg",
16     },

```

Image 6: Sample of Drink Object

4.2 Tools

Many tools and technologies were used to undertake the projects. The major tools used are described here with some explanations about the utility of each of them.

4.2.1 Visual Studio Code

Visual Studio Code is a code editor with many powerful features built for web with support for NodeJs and other JS library support to write JavaScript code along with other languages like TypeScript, Python, Java etc. VS code is available for different platforms like Windows, macOS, Linux. It provides the ability to install extensions for different languages, themes, debuggers etc for increased ease and utility. It has extensive shortcuts and is constantly adding new abilities and features like built-in Git commands, IntelliSense, Deployment. It can also be customized to meet the needs of the coder. Most of the coding was done using VS code utilizing its convenient features like code completion and code analysis[11].

4.2.2 Android Studio

Android Studio is the official IDE for Android application development and it help streamline the development process by allowing pushing changes to the app and seeing the changes immediately without building or restarting the app constantly. It also offers other vital assistance like advanced code completion, refactoring and code analysis. It comes with emulator which makes testing the application on different devices easy and quick. Some development codes were written and modified in the Android Studio and its emulator AVD was heavily used during the development[16].

4.2.3 Git and GitHub

Git is an open-source version control system used widely to manage and edit the source code. Git allows convenient branching out of code base according to roles or features and just as easy merging and deletion of the branches making working in a large group and divided projects easier than ever. It allows all the work to be done locally allowing it to be fast and more convenient. In addition to these, Git has other many features setting it apart from other SCM systems out there and git is free for anyone to use[18].

GitHub is the world's largest and most popular software development platform. It allows the developer to push the Git repository to remote server where it can be saved as private or public and pulled when necessary. It also allows collaboration, review of the code and development for developers from all around the world remotely. Moreover, it allows organizations to assign roles, organize teams, manage access to repositories and much more[19].

Git and GitHub were used to manage the code base in the project. The project did not have collaborators thus the full utilization of Git and GitHub functionalities were not necessary.

4.2.4 Trello

Trello is tool to collaborate and manage tasks for a team. Trello allows to create boards with different lists where cards can be created with attachments and description and checklist. The concept behind Trello is a simple whiteboard used in offices with sticky notes to keep track of the ongoing tasks and progress tracking. Trello also allows tasks to be assigned to members and label them according to categories. Trello was used in the project to keep track of all the research materials, progress steps, needed features and to do lists. It was also used as a tool for company members to oversee the progress steps of the project[20].

4.2.5 ADB

ADB is a command-line tool which among other things can be used during development to install, run and communicate with the Android device from the development device. The android device is considered client, and the development device a server. ADB has other purposes like access the Unix shell of a device where variety of Unix commands can be run. ADB first start the connection to the client using a port and when the connection is made it can function normally as per need. During the development process, ADB was used to establish the connection between the machine and the development server. ADB allows wireless connection to the Android device and to use ADB the

developer options must be “turned on” on the Android device. Android studio has a comprehensive guide on using ADB available on the official website[21].

4.3 Implementation

This section would explain briefly what solutions were used and possible reasoning behind it with some experience during the process.

4.3.1 React Native

React Native is JavaScript library for developing native rendering application for both iOS and Android. During the project, React Native was chosen for its simplicity for building the native apps and ease of building applications for different platforms. The React Native was used to render the UI of the application. React Router DOM was used for navigation and React Native Elements were used for components.

4.3.2 Redux and React-Redux

Redux is a state management tool for JavaScript applications. An application has multiple states and data to be stored to function properly and all those states and data belong to different views or components of the application but depend on each other to create one smooth user experience. Thus, managing all the states and data conveniently is a must which is simplified by Redux. Redux creates a store where all the states of the application is stored and all the views and components can access the state when needed. “React-Redux is the official React UI binding layer for Redux” according to the official React-Redux website. React Redux helps React to render and update the UI or components when the state in the store is updated. It checks for the updates in state and renders the change in UI without having to do that separately[12][13] .

React-Redux was used in the application to store all the data in the application using the information about the drink, user, or shakers etc on the front-end application.

4.3.3 Android Native Modules

While developing an application with React Native, it should be kept in mind that running into situations where the functionalities of React Native are not enough to accomplish a task and developer might need access to the native API and native classes of the device. An Android Native Module does exactly that, it accesses the native class and exposes it as a JavaScript class for React Native to use it. This is not limited to native classes of Android but external JAVA library used in the development which can be utilized using JavaScript. During the development, the project initially required to use external JAVA library to make an external RFID reader function which was turned into a module to be used through React Native. Later during the development, the type of RFID reader was changed and thus the module was developed which utilized the functionality of Android to listen to keyboard[14][15].

4.3.4 Other NPM packages

Using React Native for development opens the possibility to utilize the variety of packages and libraries available through npm. The project used such packages to accomplish several things. The project used Formik to handle the form, React Native Vector Icons to render desired icons. But specifically, one of the most important task of communicating with back of the machine using UART port was accomplished using external package by Basten Gao. The package helped to specify the communication details like baud rate, data bits, parity bits etc and to establish two-way communication between the android and the processor on the back.

5 Outcome

This section would describe the outcome of the project following the development.

5.1 User Application

The application developed as the project was left unfinished due to change in business decision in the company. Since, the project was designed to fit a specific machine for a specific business purpose, it was deemed fruitless to put more time and effort into the application after it was scrapped by the company. However, the application had progressed nicely where the basic features were functional, and the development was streamlined.

5.1.1 Developed Features

User Application has the ability to:

- Work with the hardware of the machine as desired
- Fetch the information about the drinks and render as needed
- User Registration using RFID tags in the protein shakers
- Report some of the transaction

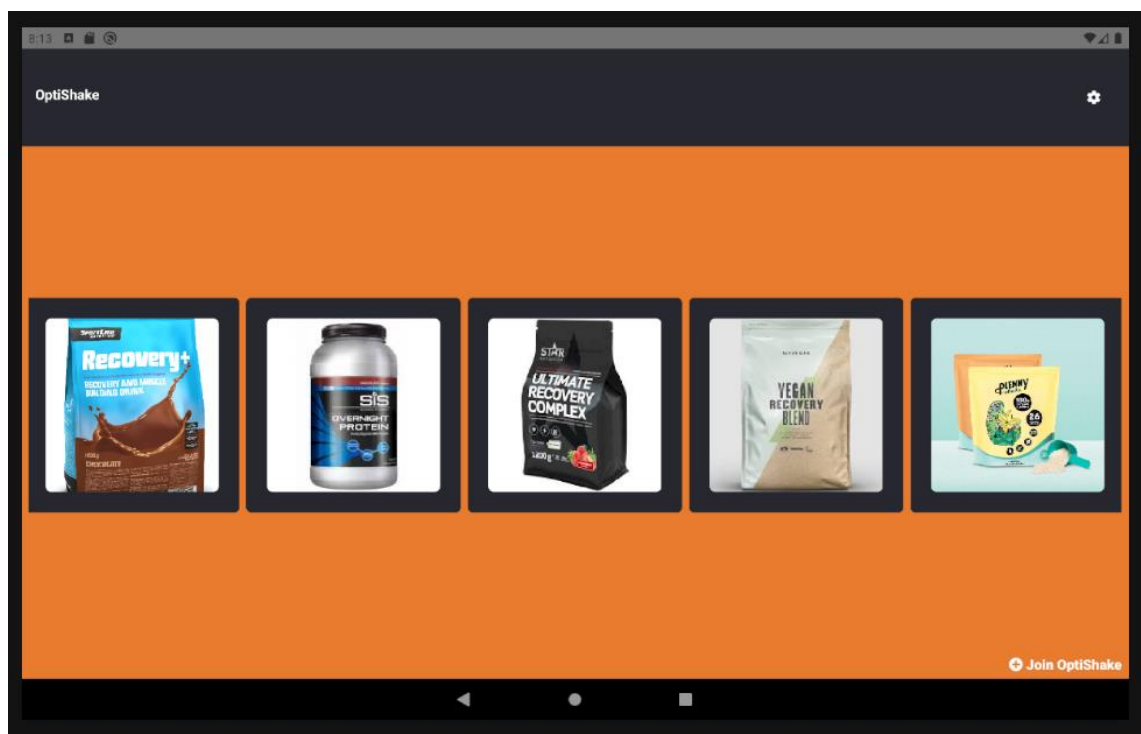


Image 7: Home Page of MVP

Image 7 presents the home page of the application where different drinks are shown in the home page which the user can click to order. And the user has not scanned the shaker yet, the application would ask the user to scan the shaker, else the info about the drink would be displayed where user can confirm the order.

Image 8: Register Page of MVP

Image 8 shows the registration page with input fields for full name, email and contact number are required. When a new shaker is scanned, the check box would show the scanned shaker.

5.2 Difficulties

As expected at the start of the project there were many difficulties faced during the development of the project. At first the lack of proper information about the machine made the start of the development process slow. The guide materials and helps was scarce due to React Native being such a new technology. Additionally, the use case was rare and unorthodox way to approach the problem so help from prior projects was scarce. Unlike React, React Native lacks many UI components in the React Native Elements library which created the necessity to make use external npm packages or custom-built components. During the

development, it was mandatory to know Java to develop custom module to control hardware through React Native, which is less than optimal solution given that the aim of React Native is to rid the necessity to program in the different languages.

There were other practical difficulties like difficulties reading guides and provided JAVA library for first RFID reader used in the project due to the fact that many variables and comments were in Chinese. The project was a solo project without any help and guidance from senior developers which might have affected the efficiency and performance to some extent.

5.3 Future of the Project

Given the fact that the project was cut short due to aforementioned difficulties and other business decision changes from the company, the project is being rethought and developed in Java. The decision to use Java mostly came with the realization that in such project, using Java is inevitable and lack of necessity to develop same application for web or iOS.

The project cannot be utilized for general purpose or by general population because it is developed for a specific machine with specific requirements and operating principles. As for the case company, the current project is scrapped.

6 Conclusion

The project was started with two main questions regarding the ability of React Native to perform when hardware is involved and the ease in development. After considering the work process and the outcome of the project, it is safe to conclude that with some extra work and with help of JAVA modules React Native can be used to make applications which can interact with hardware as well but currently this would not make the development process faster or easier given the lack of prior works and lack of resources. Although, as React Native is an emerging technology with more users and developers using it everyday, it has a real promise to be developed as library to develop such applications and more.

References

1. Bellis M. The Incredible Reason the Vending Machine Was Created [Internet]. ThoughtCo. 2019 [cited 18 April 2021]. Available from: <https://www.thoughtco.com/the-history-of-vending-machines-1992599>
2. Thomas C. The History of Coffee Vending Machines | Vista Rental [Internet]. Vista Rental. 2020 [cited 6 May 2021]. Available from: <https://vistarental.co.uk/the-history-of-coffee-vending-machines/>
3. Danielsson W. React Native Application Development [Graduate]. Linköpings Universitet; 2016.
4. Felker D. Android application development for dummies (and web access). John Wiley & Sons; 2010.
5. Operating System Market Share Worldwide | StatCounter Global Stats [Internet]. StatCounter Global Stats. 2021 [cited 6 May 2021]. Available from: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
6. Fedosejev A. React.js Essentials. Packt Publishing; 2015.
7. Hello World – React [Internet]. Reactjs.org. 2021 [cited 6 May 2021]. Available from: <https://reactjs.org/docs/hello-world.html>
8. facebook/react [Internet]. GitHub. 2021 [cited 6 May 2021]. Available from: <https://github.com/facebook/react>
9. Eisenman B. Learning React Native. CA: O'Reilly Media; 2016.
10. Boduch A. React and React Native - Third Edition. Birmingham: Packt; 2017.
11. Why Visual Studio Code? [Internet]. Code.visualstudio.com. 2021 [cited 6 May 2021]. Available from: <https://code.visualstudio.com/docs/editor/whyvscode>
12. Getting Started with Redux | Redux [Internet]. Redux.js.org. 2021 [cited 6 May 2021]. Available from: <https://redux.js.org/introduction/getting-started>
13. Why Use React Redux [Internet]. React-Redux.js.org. 2021 [cited 6 May 2021]. Available from: <https://react-redux.js.org/introduction/why-use-react-redux>

14. Native Modules Intro · React Native [Internet]. Reactnative.dev. 2021 [cited 6 May 2021]. Available from: <https://reactnative.dev/docs/native-modules-intro>
15. Android Native Modules · React Native [Internet]. Reactnative.dev. 2021 [cited 6 May 2021]. Available from: <https://reactnative.dev/docs/native-modules-android>
16. Meet Android Studio [Internet]. Android.com. 2021 [cited 6 May 2021]. Available from: <https://developer.android.com/studio/intro>
17. React Native · Learn once, write anywhere [Internet]. Reactnative.dev. 2021 [cited 6 May 2021]. Available from: <https://reactnative.dev/>
18. About - Git [Internet]. Git-scm.com. 2021 [cited 6 May 2021]. Available from: <https://git-scm.com/about>
19. Organizations and teams - GitHub Docs [Internet]. Docs.github.com. 2021 [cited 6 May 2021]. Available from: <https://docs.github.com/en/organizations>
20. What is Trello? - Trello Help [Internet]. Help.trello.com. 2021 [cited 6 May 2021]. Available from: <https://help.trello.com/article/708-what-is-trello#:~:text=Trello%20is%20a%20collaboration%20tool,for%20you%20and%20your%20team>
21. [Internet]. Android.com. 2021 [cited 6 May 2021]. Available from: <https://developer.android.com/studio/command-line/adb>

