Expertise
and insight
for the future

Nam Anh Nguyen Le

Using React Native in warehouse management application

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

24th September 2019

Author Title Number of Pages Date	Nam Anh Nguyen Le Using React Native in warehouse management application 41 pages 24th September 2019
Degree	Bachelor of Engineering
Degree Program	Information Technology
Professional Major	Mobile Solutions
Instructors	Kari Salo, Principal Lecturer

The topic of this report is “The usage of React Native in managing and operating the system in Logistic Industry”. This thesis is going to introduce the fundamentals of Mobile Development and Logistic Industry. Furthermore, the problems and challenges in this business will be presented along with the solution when using React Native. With the new features and the capability of integration of new technologies, React Native can be used effectively in improving warehouse management systems performance.

Inheriting from Javascript, one of the most well-known programming languages, React Native was developed by Facebook company for implementing cross-platform application in both Android and iOS devices. React Native has a similar concept with React, the web development tool; but instead of using web components for developing the user interfaces, React Native uses the built-in native components for the mobile platforms. After being released in 2015, both React and React Native have gained popularity among software and mobile community, and they are being used in a wide range of areas of daily life, especially in Logistics and Warehouse Industry.

A mobile application is going to be developed and launched using React Native for idea.invest, a Logistics and Software Solutions company in Hamburg, Germany.

Keywords	React, React Native, Mobile Application, Warehouse, WMS

Contents

List of Abbreviations

1 Introduction 1

2 Fundamental of Warehouse Management System 4

2.1 Warehousing Processes in Logistic Industry 4

2.2 Evolution of Warehouse Management Systems 6

3 Developing Mobile Application with React Native 13

3.1 Fundamental of React/React Native 13

3.2 Components Lifecycle 14

3.3 JSX and Styling 16

3.4 Handling and Managing States and Props 17

4 Using React Native in Developing Native Application 20

4.1 Advangates 20

4.2 Disadvantages 22

5 Case Study: Building a Mobile Scanning

Application for Warehouse Systems 23

5.1 Project requirement analysis 23

5.2 Project design model 25

5.3 Project initial setup 28

5.4 Project implementation 30

5.5 Project discussion and summary 37

6 Conclusion 38

References 40

List of Abbreviations

CSS	Cascading Style Sheets. A stylesheet language describes how elements should be rendered on the screens.
API	Application Programming Interface. A set of subroutine definitions, protocols and tools for building application software.
REST API	Representational State Transfer API.
UI	User Interface
NPM	Node Package Manager
IDE	Integrated Development Environment
JSX	Javascript XML. An extension to the JavaScript language syntax which is similar to HTML
Avis/Advice	The document which holds the information for the orders.
WMS	Warehouse Management System.
CPU	Central Processing Unit. It is the unit which performs most of the processing inside a computer
MDE	Mobile Datenerfassung in German. It is the name of the new scanning mobile application used in the warehouse

1. Introduction

The purpose of this thesis is to demonstrate the usage of React Native in developing mobile application for warehouse management system. In detail, it is going to provide the basic information about logistic industry and warehouse management. The challenges and obstacles throughout the evolution period of this field will be demonstrated with some exampled systems. Moreover, the launching of new technologies has been designed and released to meet the needs of high complexity warehouse management systems and mobile application, in this case is React Native library.

React Native is an open-source mobile application framework designed and released by Facebook in March 26th, 2015. It is written in Javascript, one of the most popular programming languages for the past ten years. According to a Stack Overflow Developer survey in 2017, Javascript was on the top among the other programming languages such as Java, C# or Python, as described in figure. 1.[1] It also has been used in many kinds of software and application, as the basic requirements for most of the job offers, which is stated and concluded in figure 2, an online platform for searching work placements.[2]

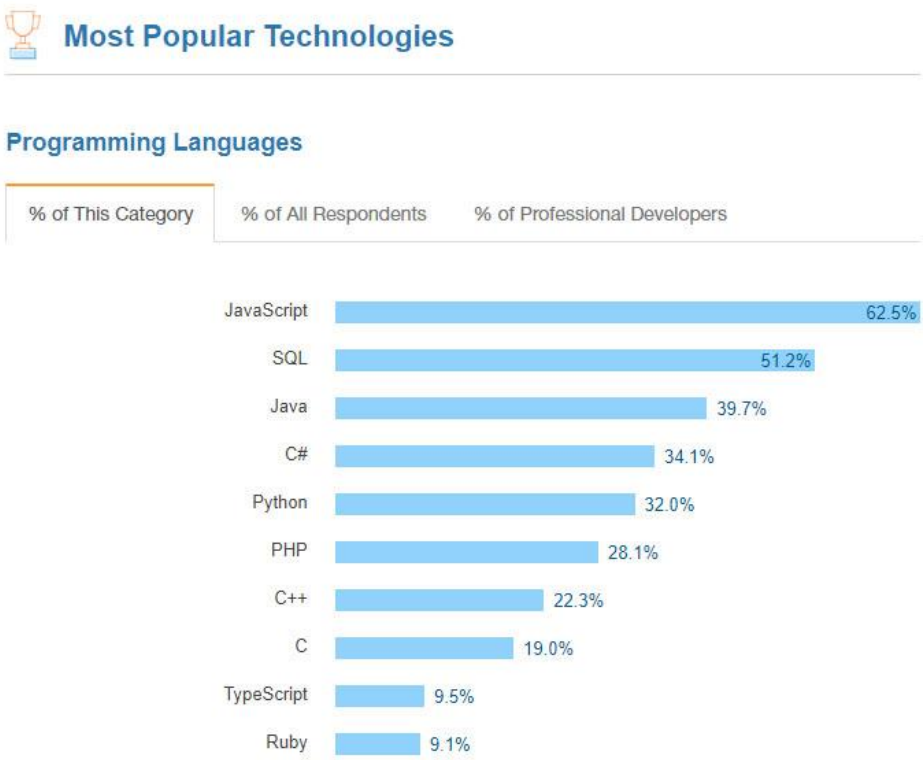

Figure 1 Top 10 most popular programming language in 2017, according to Stack Overflow

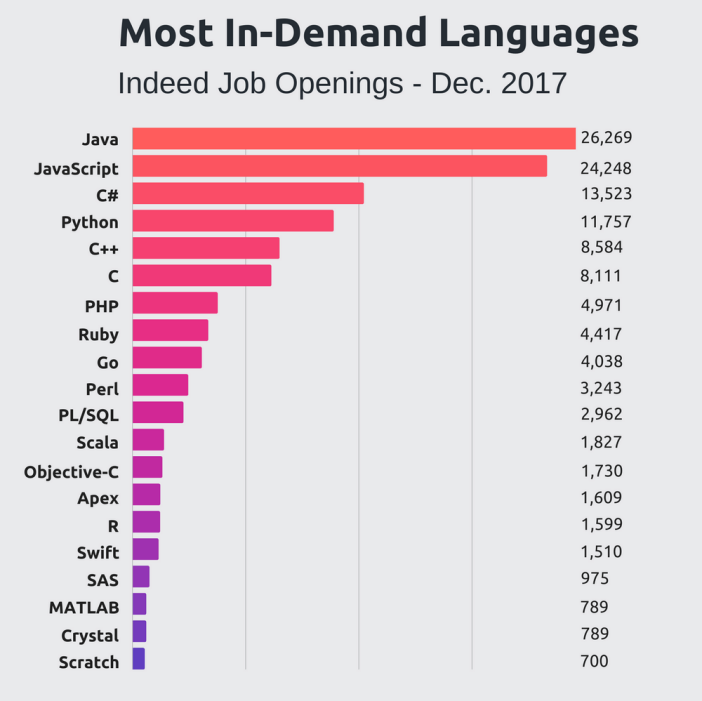

Figure 2 Top programming languages in job requirement in 2017, according to Indeed. [2]

This thesis also focuses on developing and maintaining the efficiency of warehouse management systems. This software operation is built for controlling and tracking the entire warehouse procedure by storing and extracting the data of the internal and external products within the facilities and from the clients. Due to the high demands worldwide, warehousing market is also growing rapidly every year for decades. According to Zion Research Analysis in 2017 in the below figure, the global warehouse management systems market increased more than $600 million, from $1.37 billion in 2016 to around $2 billion in 2019, and it is forecasted that would reach $3.04 billion in 2022. The growing of warehouse market can lead to the overload and un-efficiency of the systems, which need to apply the new technologies in order to overcome the warehouse obstacles.[3]

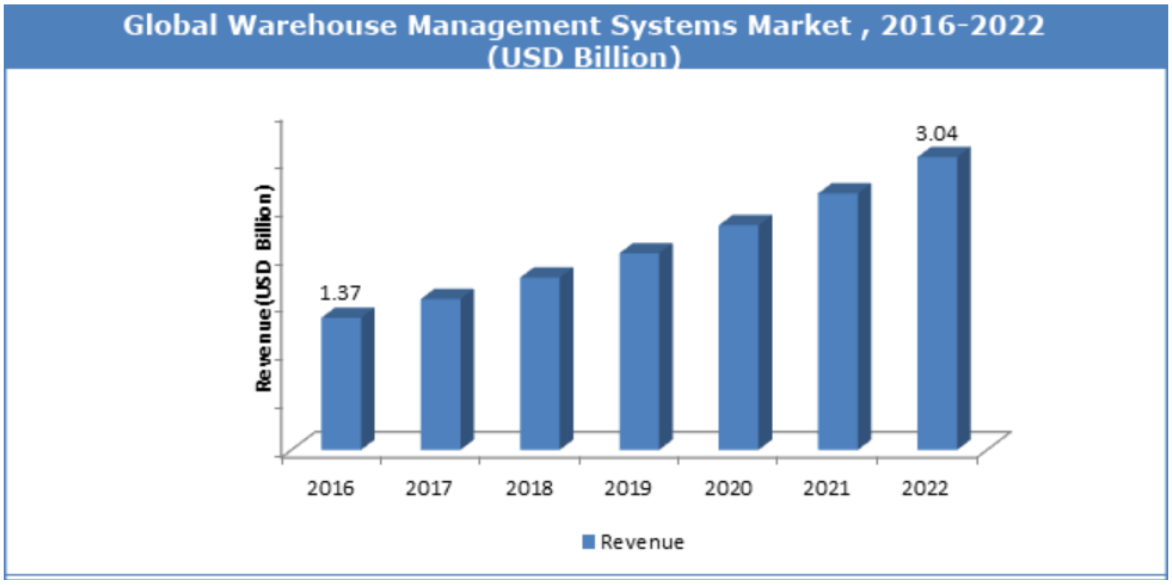

Figure 3 The worldwide expansion of warehouse management systems market in the period of 2016 – 2022

Because of the continuous growth of React Native and the relevant advanced technologies, the developer community can depend on them as the great tools for designing mobile applications. React Native provides a reliable and friendly native environment along with the high performance and flexibility of integration with pre-built web components which open the opportunity for developing and enhancing warehouse systems.

2. Fundamental of Warehouse Management System

2.1 Warehouse Processes in Logistic Industry

For many years, logistics has become a major part of supply chain management, including planning, implementing, controlling the transport and storage of goods and services as well as relevant information from the origin to the destination in order to meet the customer requirements. Activities of basic logistics management include importing and exporting freight management, fleeting management, warehousing, ordering fulfillment, designing logistics network, inventory management, planning supply demand, third service provider administration. At different levels, the functions of logistics also include finding sources of inputs, planning production, packaging, managing customer service. Therefore, logistics management is an integrated function that combines and optimizes all logistics activities as well as integrates logistics operations with other functions such as marketing, business, production, finance and information technology.

Warehouse is a type of logistics facility used for receiving, storing and preserving goods and products of enterprises for final distribution to consumers or other businesses. Hence, warehouse management is the process of operating a warehouse and distribution system in an efficient way. In order to achieve it, the system follows the logistics principles called “6Rs”, which focuses on the targets: “The right goods have to be at the right time, in the right quantity and the right quality, at the right location with the right cost”.[4]

Based on the logistics framework, the typical basic processes and functions of the warehouse management system are described in the figure below.

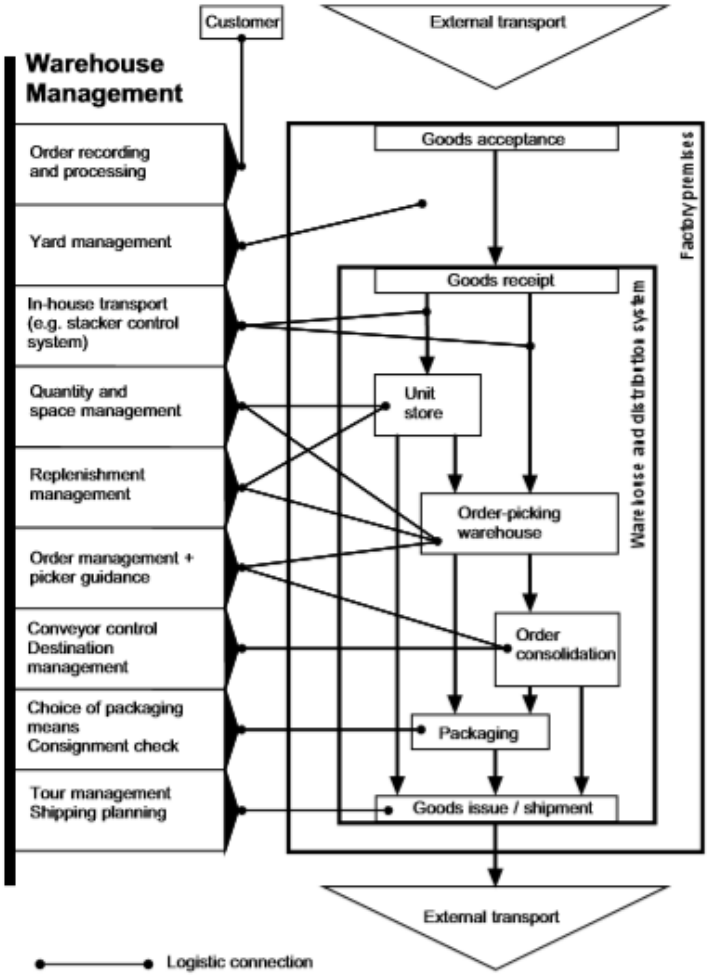

Figure. 4 A typical warehouse management processes.

The warehouse procedure starts with handling goods acceptance and receipt, after they have been ordered by the company's dispatcher and sent to the depot. The notification of the new incoming goods will be received with the relevant crucial information about the sender, the articles or products such as names, identifiers, quantities and cost, and the delivery date. The incoming goods will be checked by comparing data between this notification and the bill of landing, as a pre-step of the WMS workflow. Furthermore, based on this notification data, the goods receipt will be generated for the purpose of inspecting the new products. Warehouse staff uses this e-document to check the actual quantity and quality of the incoming goods in the specific containers, and then report this final result to the admins.[4]

After checking and accepting the new goods from the external source, these articles will be distributed to the warehouse facility. Each one is stored in a specific container or pallet attached to the related required information such as: identification numbers of the articles

and the location, quantity and client of the article. This approach brings an advantage for the picking and retrieving process. For instance, the present orders have to be checked with regard to their feasibility; in the actual case is the combination of the right products and their relevant quantities. Any inappropriate incidents will be reported to the admin for keeping the consistency in the warehouse. [4]

The last two steps in the warehouse procedure are packing and shipping, in which the picked products are handled before being delivered to the clients. After being selected from the internal storage, these articles will be sent to the packaging department, where the checking process happens additionally in order to verify the required information and status of them. In case everything follows the correct order, these items will be wrapped in the appropriate cartons and then allocated in different containers. The warehouse workers are responsible for scanning those containers labels and transfer them to the transport vehicles which take all these products to the shipping stage via cargo vessels or aircraft.

2.2 Evolution of Warehouse Management System

Warehouse management system has been a crucial part of supply chain procedure and logistics industry for ages. The ancient Egyptian were using the concept of warehousing when they sought to manage gathering grains, keeping on records on the granaries in storing and taking out all the products. This approach is the basic concept of inventory or warehouse management in the modern world. These below paragraphs are going to introduce the growth of WMS since the middle of 20th century.

The evolution of warehouse management has changed substantially, with the following shift of new technologies and high demands during the 1960s. Despite of using only railways, a new trend was introduced using trucks in moving freight services, which led to the formation of the National Council of Physical Distribution Management in 1963 to control and focus on the warehousing, material handling and freight transportation. Accordingly, the academic research and education also put attention on these fields to meet the industry's growing recognition of demand in the warehouse management. This area has been more widely recognized in both industry and academia because largely the fundamental model change occurred in the 1960s and 1970s involving computers. Before the 1960s, almost all transactions and record keeping were done manually by hand. However, the presence and developing of computer during this period has brought a great improvement in warehouse management in optimizing stored data and shipment routing.

In 1967, IBM developers released the very first computerized inventory management and forecasting system, which called IMS; and it was installed and used by NASA to keep track on all the data. According to Don Lundberg, one of the systems engineers developing this system, the IMS has become a necessary part in many of the largest corporations in the world in operating their daily business. [5]

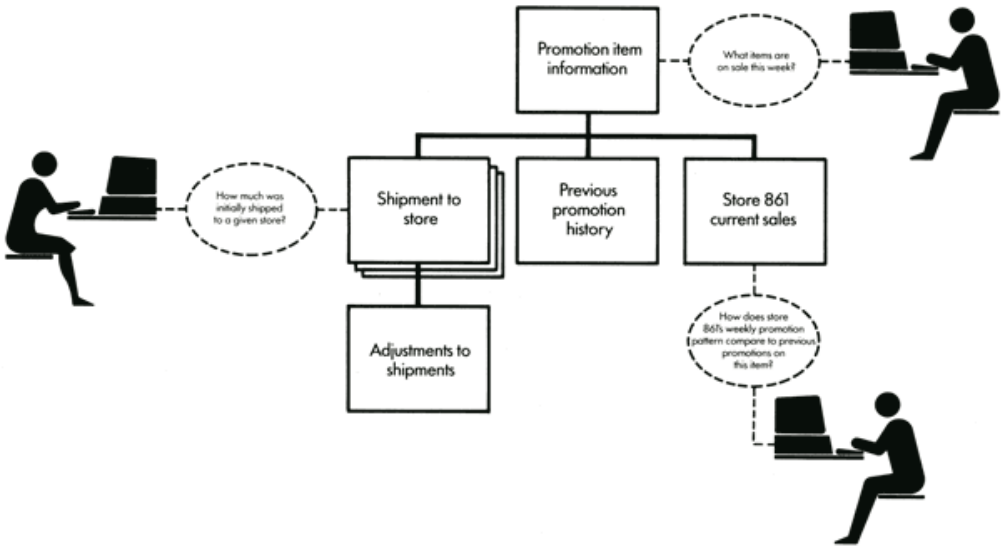

Figure. 5 This diagram illustrates how IMS makes information available to its users.

In the periods of 1980s and early 1990s, the world witnessed the arrivals of new technologies using in warehouse management system. These included UNIX based operating systems, relational database engines, standardized networking protocols, client /server architecture, graphical based user interface (GUI), new bar code scanners, and improved printers. Especially with the usage of relational databases, warehouse companies were able to retrieve and handle with goods data in an efficient way. Furthermore, most of the products information could be verified by using the bar code scanners, which guaranteed the consistency in the data management system.[6]

From the middle of the 1990s to early 2000s, the logistics industry and warehousing have been expanding dramatically, together with the growth of manufacturing all over the world, especially in America and China. The trading market between these two countries and the others has been increasing drastically in this period, according to Global Trade Atlas. China has become a major part in the field for exporting and importing world commodities or primary goods. In the period of 7 years from 1999, the total number of imported machinery products to China increased from $63.1 billion to $271.3 billion. This high demand has led to the expansion of the warehouse facilities and structures, beside of the large number of employees. As a result, the evolution of the warehouse management

system was inevitable in order to meet the increased demand and high amount of data, avoid overstocking and reduce procedure cost. [7]

	1999	2000	2001	2002	2003	2004	2005
Electrical Machinery	35.3	50.7	55.9	73.3	104.0	142.1	174.9
Machinery	27.8	34.4	40.6	52.2	71.6	91.5	96.4
Mineral Fuel, Oil, etc.	8.9	20.7	17.5	19.3	29.3	48.0	64.2
Optics, Medical. Instr.	5.0	7.3	9.8	13.5	25.1	40.1	49.9
Plastic	11.6	14.5	15.3	17.4	21.0	28.0	33.3
Organic Chemicals	5.5	8.3	9.0	11.2	16.0	23.8	28.0
Iron and Steel	7.2	9.6	10.9	13.2	22.2	23.6	26.2
Ores, Slag, Ash	2.2	3.1	4.2	4.3	7.2	17.3	25.9
Copper & Articles Thereof	3.1	4.7	4.9	5.7	7.2	10.5	12.9
Vehicles, Not Railway	2.4	3.6	4.5	6.5	11.8	12.9	12.2
Misc. Grain, Seeds, Fruit	1.6	3.1	3.3	2.8	5.7	7.3	8.1
Cotton and Yarn, Fabric	2.4	2.8	2.9	3.3	4.7	6.9	7.0
Aircraft, Spacecraft	3.2	2.2	4.4	4.1	4.5	4.9	6.6
Paper, Paperboard	1.6	2.6	2.7	2.9	3.9	5.2	6.3
Misc. Chemical Products	2.2	2.5	2.6	3.8	4.9	5.1	6.0
Wood, Articles of Wood	2.9	3.7	3.5	4.1	4.6	5.2	5.7

Source: Global Trade Atlas using Chinese data.

U.S. Department of Commerce, International Trade Commission; Global Trade Atlas; International Monetary Fund, Direction of Trade Statistics Quarterly, June 2006.

Figure 6. China's Imports by Major Commodity, 1999-2005

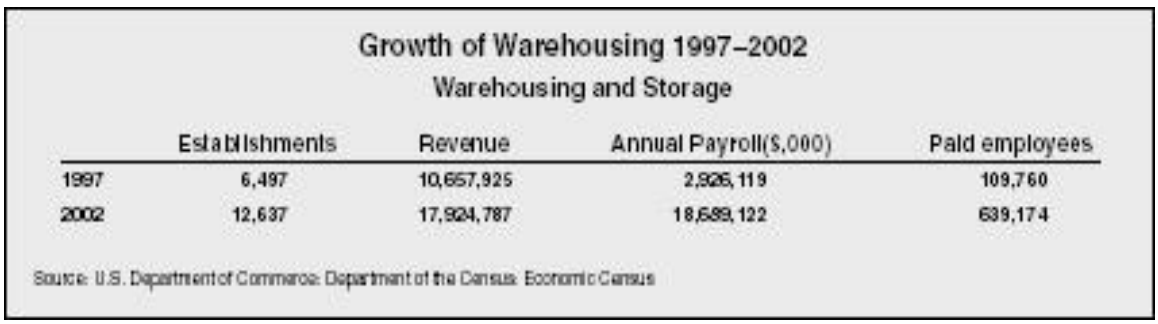

Figure 7. The growing of warehousing from the period between 1997 to 2002

Source: U.S. Department of Commerce: Department of the Census: Economic Census

After the middle of the 2000s, the WMS has moved to the new milestone relying on the new technologies. For instance, Amazon has been one of the major parts in the logistics industry which providing a significant amount of worldwide commerce products. This company developed and released Amazon Elastic Computer Could, and then Amazon Web Services. The other software companies such as Microsoft or IBM have also launched and offered their new services using cloud computing and internet of thing. Furthermore, a new technology trend called software as a service (SaaS) had been used in many software and logistics companies which moved their operations to the cloud and managed WMS online. This has opened a new era of developing management system applications in order to solve the existing warehouse obstacles and optimize its procedure.

During this period, SAP warehouse management systems have been developed and launched by SAP SE, a German multinational software corporation, as one of the most advanced software for providing the solutions for supply chain management. The extended version, named as SAP EWM, of this application has been using in many companies as the standard systems in their facilities. EWM is designed for managing and handling complex logistics process efficiently, which is including goods receipt and goods issue, complex cross-docking, slotting, packing and shipping procedures, as well as cross-function activities such as LSP (Logistics Service Provider) integration, labor management and analytics. According to a report in Enlyft database, there have been more than 1.529 companies that have been using SAP EWM as the main technology to operate their key activities within the systems. America, one of the top market and industry area in the world, has the most number of enterprises using this application, with around 44% over the total number of customers. [8]

Figure 8. A typical WMS used by Seldat Technology Services.

One of the popular warehouse management systems at the moment is WMS360, developed by Seldat Technology Services. This application supports for both inbound and outbound planning process, which includes appointment scheduling, picking, packing, shipping and other relevant actions inside the warehouse. By using Amazon Web Service for cloud hosting, WMS360 can manage to operate different services within system network. This brings advantages for the logistics company dealing with high amount of data and complex warehouse activities. For instance, Century, a Vietnamese manufacturer, works and produces high-quality polyester filament yarn and was in the top 500 largest companies in Vietnam, according to VNR500. It has had large warehouse facilities where were storing their products, and one of them use a high complexity racking system to track inventory throughout the entire process. By using and applying Seldat System, the company were able to handle tasks and product workflows effectively, improve the picking process by 25% and worker efficiency by 12%. Pham Ngoc Thai, the IT manager of Century, stated that this approach was the suitable one for his company which brought many positive results, and they wanted to expand the system by continuing cooperating with Seldat Technology. [9]

Besides the Seldat system, nowadays there are other popular mobile applications have been used by warehouse users, such as Snappi Warehouse Inventory and Shipment, LoMag Warehouse Management, Inventory Management. All of these software support systems for scanning barcode, which is the basic requirement for handling the warehouse procedure in tracking goods shipped in and out. Moreover, they allow user to create documents and print them out in various formats.[10] However, there are some existing limitations within those modern applications features. One of the obvious obstacles is that not all of these services support for both iOS and Android systems. With those one having two versions for different mobile platforms, developers have to implement them separately in Swift or Objective-C for iOS, and Java or Kotlin for Android with lots of effort and time consuming.

Figure. 9 LoMag warehouse application with unfriendly mobile view.

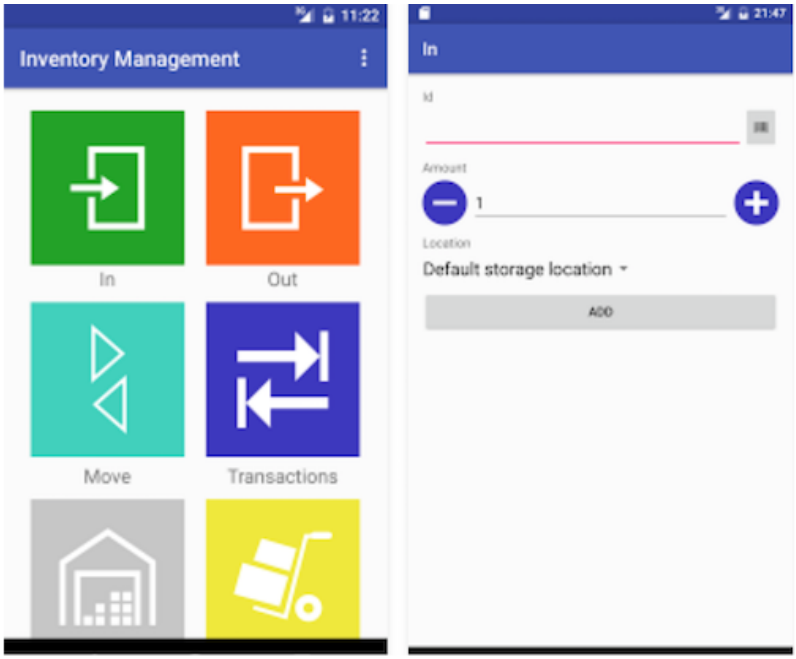

Figure. 10 Inventory Management warehouse application developed by Industrialit, a Netherland software company, has a simple and meaningful visualization in Android platform

3. Developing Mobile Application with React Native

3.1 Fundamental of React/React Native

React Native is an open source Javascript framework which was developed by Facebook company and is used for implementing native mobile application in iOS and Android. Both React and React Native are written in Javascript, inheriting the concepts of the programming language in order to build user interfaces, but in different platforms.

The process of rendering and presenting the frontend components in the screens is working in different layers. In core Javascript, developers can change and render the interactive user interfaces in a browser via DOM. By accessing the specific HTML elements, programmers can change these documents style, structure and content. In React and React Native, instead of using normal DOM, they use Virtual DOM as a layer between the code and the actual rendered presentation in web browser or mobile screens. [11]

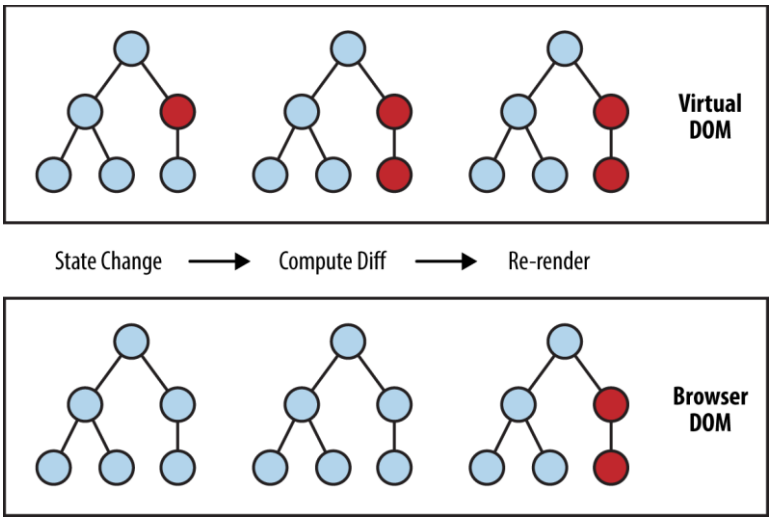

Figure. 11 Comparing the Browser DOM and the Virtual DOM

This Virtual DOM has been become an important part of rendering in React and React Native. First of all, it improves the performance of these two frameworks during implementing and compiling. Instead of editing browser's DOM and rendering directly the changes, they use an in-memory version of the DOM to calculate the differences in order to minimize the amount of data to be re-rendered. Secondly, using this Virtual DOM allows React or React Native invoking Objective C API or Java API in order to render the native iOS components or Android components. [11]

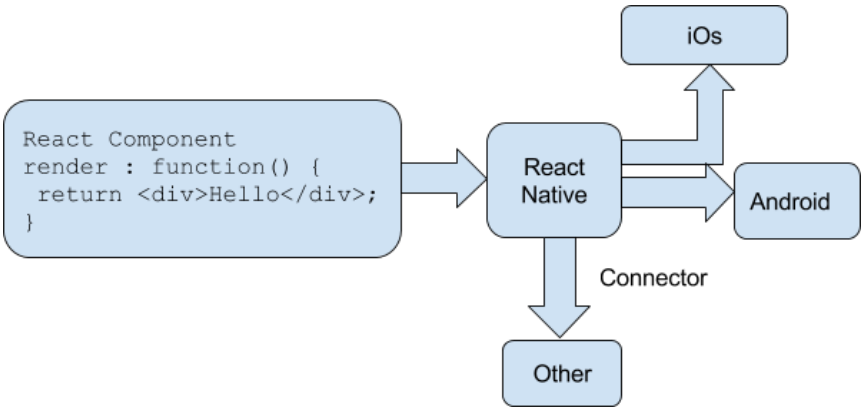

Figure 12 React Native can render to different mobile platforms

3.2 Components Lifecycle

The essential concept when creating a new React Native component is its lifecycle. This concept is similar in native iOS and Android development, when the Activity and UIViewController have their own separate phases during compiling and running time. The lifecycle functions will be automatically called by the system at the appropriate time, and developer can rely on that to launch the necessary processes. React Native handles the Virtual DOM updates through manipulating states and props in three stages, which are named as mounting, updating and unmounting. Mounting describes the process of a component being initialized and rendered in the DOM. On the other hand, unmounting is when a component is removed from the DOM. There is also a stage of updating between mounting and unmounting, where the data can be updated.[12]

The first main lifecycle flow of a React component is mounting, where the component is being initially rendered. The method "constructor()" will be called in the first place with

props parameter to declare the values of the states and reach out the props values which are passed from other components. The next important lifecycle method is “render()”, which returns React Native element created with JSX. [11]

Once mounting is complete, the updating cycle can be called when a change happens with the props and state. The common method which is used for handling the updates is “componentDidMount()”. Developer can trigger the modification of the states or other data inside the component by wrapping them in the conditions checking the specific props or states with their previous values. Other rarely used methods during this stage are “shouldComponentUpdate()” or “getDerivedStateFromProps()”.[12]

The last stage in the component lifecycle is unmounting, in which the component is removed from the virtual DOM. If developers want to get any data values or change the JSX element before the component is unmounted and destroyed, they can call the method “componentWillUnmount”.

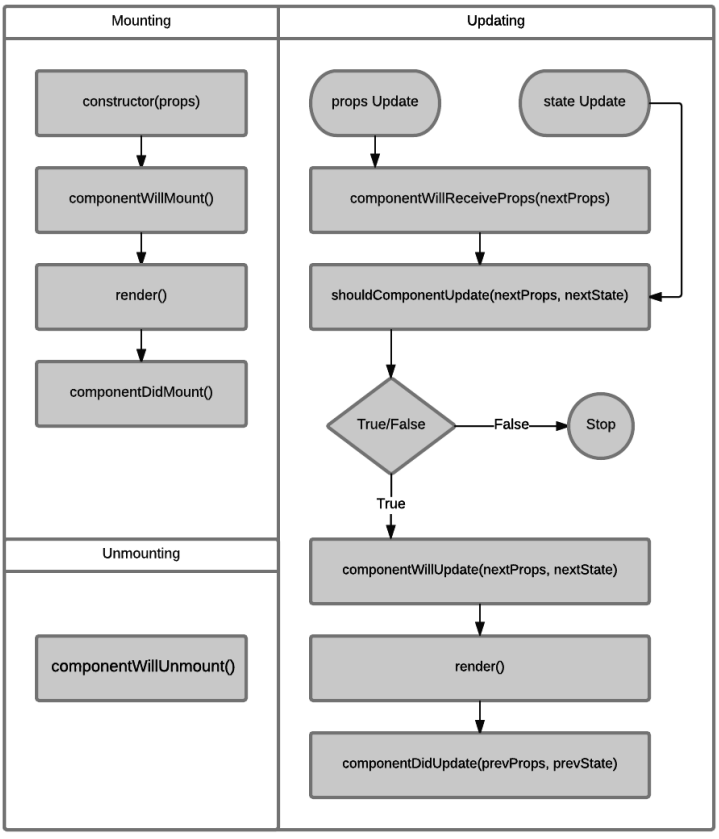

Figure 13 Component Lifecycle inside React and React Native application

3.3 JSX and Styling

In React Native, the views presented in the mobile screens will be written in JSX, a combination of declarative markup language and Javascript syntax. Different from the other similar framework such as Vue.js or Angular which is still using HTML to define the layouts, React Native use mainly JSX for this purpose with the strict rules to follow. For instance, the HTML tag can be self-closing without using a forward slash before the end; but in JSX, it is required as shown in the illustration below. [13]

HTML: <img src="my/image.jpg" >

JSX:

To beautify the layout created in JSX, React Native uses Javascript syntax to directly create style objects with the similar properties in CSS. One of the most powerful tools to be used in styling React Native component is flexbox layout system. It enables us to make a declarative, flexible and responsive layouts for different mobile platforms. Furthermore, same as HTML and CSS, React Native's style properties can be used directly inside JSX elements, or created in a separate object and will be called in the styling props of the specific parts in the layout. [13]

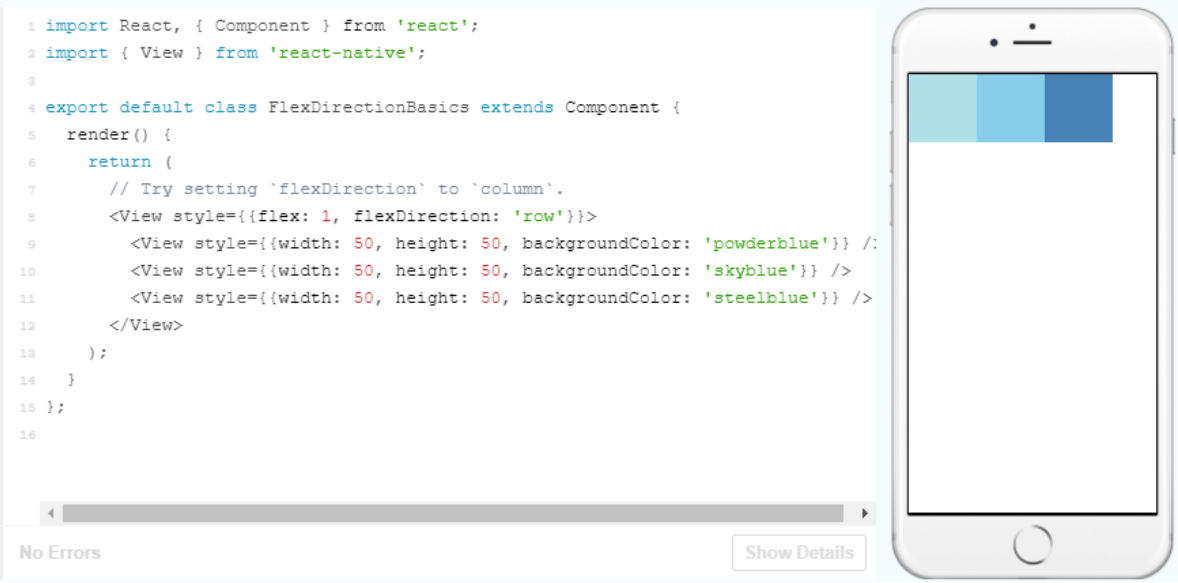

Figure. 14 Introducing Flexbox in React Native

3.4 Handling and Managing States and Props

The focus of this chapter is manipulating and managing component data, which can be handled by using props and state.

Props stands for properties, which are the immutable data passed into a component. During the implementation, props values cannot be changed after the initial render of component. The components receive props from parent component and are only allowed to read their values. If there is any update in props, React Native component has to re-render the JSX that was used to render it in the first place.

```
// Parent
export default class ScreenOne extends React.Component {
  render () {
    return (
     <View>
       <Heading message={'Custom Heading for Screen One'}/>
     </View>
    )
  }
}

// Child component
export default class Heading extends React.Component {
  render () {
    return (
      <View>
        <Text>{this.props.message}</Text>
      </View>
    )
  }
}
Heading.propTypes = {
  message: PropTypes.string
}
Heading.defaultProps = {
  message: 'Heading One'
}
```

In the above example, the “ScreenOne” class has component Heading, with props as massages. This parent class will determine the value of the message, and in the above case, it is “Custom Heading for the Screen One”. The child component called Heading can be reused in many places with the different appropriate values. Furthermore, these parent classes do not need to change this value and can use the default value by setting the “defaultProps”.

Besides of using props, React Native class also uses state to represent the data and control the workflow. Different from props, state is the internal mutable data of a component, which means developer can declare the initial value of the state of a component and change it over time. This data can be handle by using the ”setState()” function inside the class component.

The preferred practice is using third party library for managing states of the entire application. One of the popular combination is using React or React Native and Redux together for this purpose. Redux has four main parts: Store, Action Creators, Reducer and the React Component. Action Creators is the place where the events inside the application are take places, and these actions are going to be dispatched to the Store object in order to update the states whenever users trigger an event in the user interface. This dispatched action information will be forwarded to the Reducer to create the new states, instead of modifying the previous ones. After the new states have been returned from the Reducer, they will be sent to the React or React Native components in which the views are going to be re-rendered.[14]

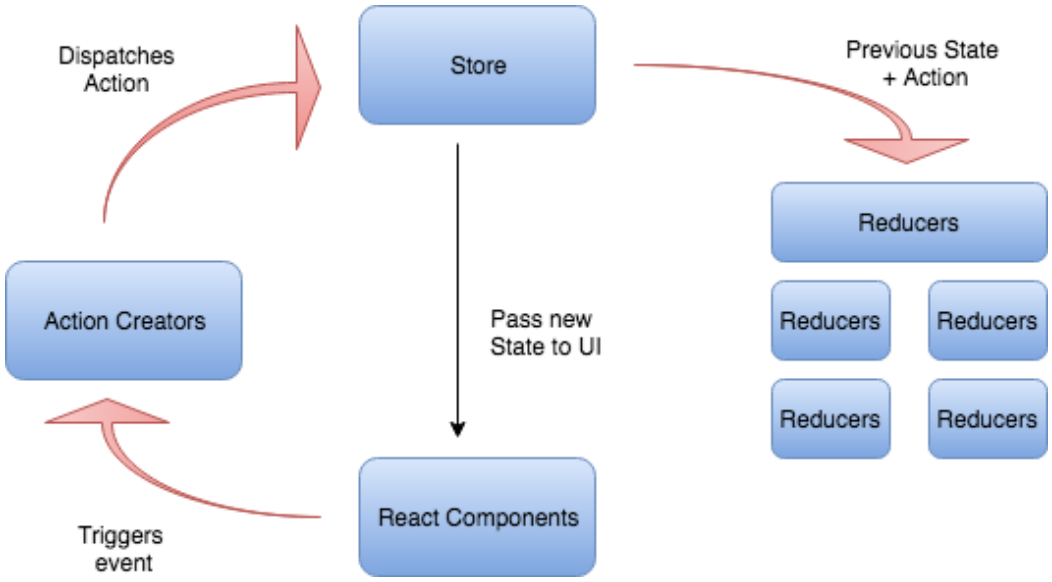

Figure 15 The workflow of managing states by using Redux in React and React Native

4. Using React Native in Developing Native Application

4.1 Advantages

This chapter is going to present the reasons why React Native has been chosen as the suitable technology for many big companies to develop their applications. Facebook, the creator company of React Native, has been using their own framework to implement the application running the mobile devices. Besides Facebook, Airbnb, Instagram and Skype use React Native in their development process. By utilizing the framework advantages, these companies have earned positive results, drew more attention from customers and expanded their market rapidly.

Native development mostly provides multiple mobile components for creating user interfaces. However, in some cases, the implementation requires other pre-built components which are not available in the program libraries. As a result, users have to create them on their own, but this approach is not easy in the native tools compared to web development. Furthermore, the complexity of this software is increased due to unfriendly interfaces and workflow, which is not practical for the usage in the warehouse. These issues can be overcome by reducing the unnecessary data and functionalities displayed on the mobile screen, using the new technologies such as Angular 6, React Native or Vue Js which are able to adapt and apply web-based components to the native mobile platform.

React Native can help developers to save time and reduce cost during the implementation phase for the warehouse industry. This Javascript library allows users to design and build their applications in both Android and iOS operating systems. Instead of writing the mobile app separately in Swift or Java, developers can use only Javascript code for their work. By this way, users can save a lot of time and money for the development tools and processes, especially when building iOS app which can only be written in Objective-C or Swift requiring Apple tools such as Xcode. Moreover, React Native provides a number of additional features such as live and hot reloading, which speeds up the visibility of changes and debugging processes significantly faster.

The second major concern about choosing the appropriate technology for implementation purpose is improving the performance of the applications. Compared to the native development, React Native has worked effectively within the devices. In order to clarify this point, a software developer named John Calderaio has been doing an experiment in

React Native and iOS. Calderaio created a simple application written in both Swift and this Javascript framework, then comparing the functionalities usage in the mobile device memory and CPU.[15]

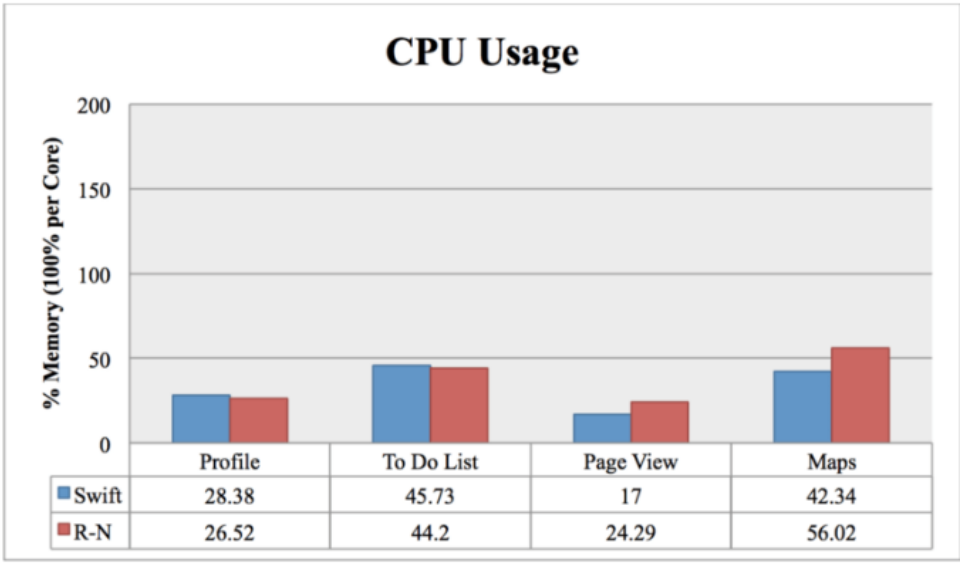

Figure. 16 The usage in CPU between React Native and Swift application

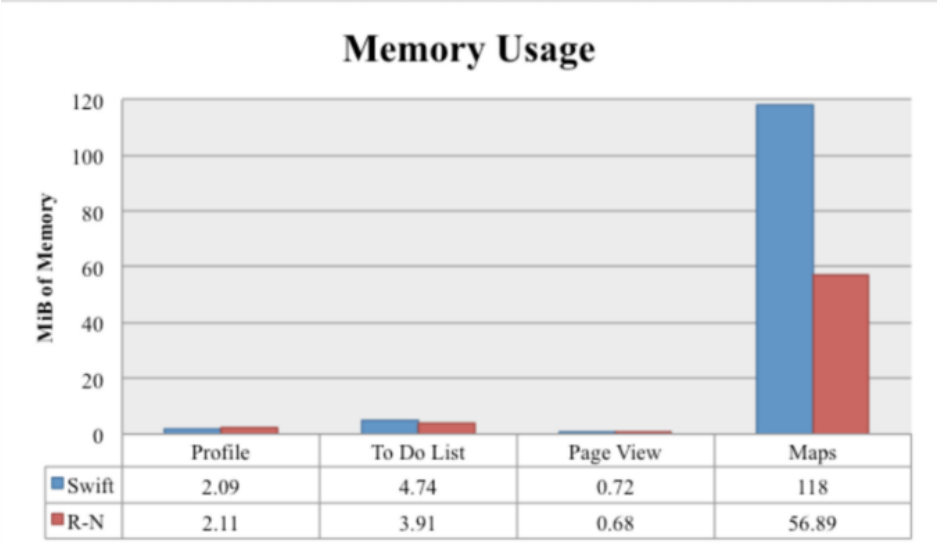

Figure. 17 The usage in memory storage between React Native and Swift application

These two above figures display the data results within the conducted application in consuming memory and CPU usage of mobile devices. It is obvious that there are not huge differences between the two versions of this software in the first three screens: Profile View, Todo List View and Page View. In the MapsView, React Native consumed

about 10% memory more than Swift in CPU usage. However, the Javascript library only used 56,89 MiB of the device storage, which is significantly lower than native iOS programming language using more than 118 MiB. Undoubtedly, using React Native can minimize the memory consumption for better performance.

4.2 Disadvantages

Besides of the advantages in which React Native can bring to the community, this new and less mature framework also has many drawbacks that can affect the development process. First of all, the React Native developer community has released a lot of different versions of this framework, which lead to the inconsistency during the integration between multiple libraries. The developers face this struggle occasionally when they want to implement and maintain their products.

Secondly, writing the code using React Native is not a native method to implement an application as in the Web or iOS, Android development. This framework can not provide the full functionalities comparing to Swift, Objective-C in iOS, or Java and Android Studio in Android. Furthermore, it is better in the native development when dealing with debugging and reviewing the code, because of the presence of the developer tool inside the implemented platforms. Despite of the usage of hot reloading and live reloading update, React Native is still lacking in high-quality experiences for the users.

One of the major concerns when using React Native for cross-platform mobile development is the compatibility of the features and user interface within the app in different platform. Android and iOS have different design principles and hardware devices, so that the developers have to spend more time and effort in writing React Native app in order to satisfy the concepts and characteristics of the particular mobile platform.

5. Case Study: Building a Mobile Scanning Application for Warehouse Systems

5.1 Project requirements analysis

The following chapter demonstrates the process of using React Native to develop a mobile application. This application, called MDE, is a part of the warehouse management system created by a German logistics company named Idea.Invest in Hamburg. MDE provides various of semi-automated methods for the workers in the warehouse procedure. By using the attached hardware camera, this software allows user to scan those identifiers of all logistics entities inside the warehouse in order to handle all the processes.

This MDE mobile application is a part of the warehouse systems, so it also handles various complex functionalities which are related to each other. Due to that reason, this main topic of this chapter is going to focus on the GoodsReceipt procedure. The other functions of the warehouse will be introduced briefly, because the workflow of different systems support my main thesis topic.

5.1.1 Functional Requirements

The very first functionality inside the GoodsReceipt is scanning and accessing the receipt generated from the admin panel, the desktop application used by the warehouse admins. By using a special scanner within the device, user can get the QR code of the goods receipt and filling it in the input field presented on the screen. As a result, the warehouse worker can access all the needed data of the corresponding receipts and use them for the next functionalities. One of the major data is the id of the client, which is needed for setting the client filter applying for the whole application.

The second step, which is also the next function of the process, is searching for the articles. These articles are fetched using the id advice property inside the above goods receipt and they will be collected from external sources. This document is automatically generated when the admins receive the orders from those customers. In case there is no advice, or the advice contains wrong information, the goods receipt is still being created

without any relevant advices. Warehouse workers are allowed to search for the internal articles inside their facilities, in which they are matched with the information from the orders. In both cases, the users can set the filter based on different key data for the specific products such as the external id or normal id of the articles.

The main function of GoodsReceipt process is to add the entities required from the order. After fetching these ordered articles from the advice or by manual search, this software allows users choosing an item from the list and adding the quantity, scanning the serial number and the container holding these articles. With this required information, user can send the request to add the new entity for this receipt.

Additional functions that are required for this process are observing the added entities and pause, complete the process. It is essential for any application that the users can track their progress, in this case is the seeing the list of the entities that were added of the corresponding receipt. Furthermore, the users should have the capabilities to pause the process whenever they want, or they can finish it by clicking on some buttons.

5.1.2 Non-Functional Requirements

Besides of those functional requirements, software developers and engineers must be aware of the non-functional requirements, which define the quality of the entire system, process. These characteristics of the application are including the friendly user interfaces, seamless and flawless user experiences, high performance and security, reliability and maintainability. Failing to meet these requirements can lead to the failure of satisfying customer's needs and the application system.

The application interface is the first factor which has to be concerned about during the implementation and practical process. In order to acquire a simple and meaningful user interface, the application design should follow a several principles specializing for the mobile devices. These principles can be found on Apple or Google websites, where they define the rules of using and presenting UI elements such as text, input field, icons, buttons… For instance, the visibility of texts and buttons must be clear and big enough for warehouse workers being able to see and interact with. Moreover, choosing the appropriate colors pattern and theme for the entire software has to be taken in account seriously.

Along with the user interface, developer must be responsible for the high-quality experience of the users, and the maintainability and the reliability of the application. Showing the errors with the meaningful text or warning pop ups, avoiding unnecessary actions or tasks are the practical and simple examples of improving user experiences. As stated in Nielsen usability principles, the application must always inform the users about the status of the process and system during the running time.[16] For instance, the loading state of the software should be presented by the spinners, or the successful results of the tasks should appear using the green check mark. Additionally, the application structure should be presented in a scientifically and logically approach, which can be attained via adopting software design patterns.

5.2 Project design model

Based on the initial requirements presented above, the designs of the entire systems and especially the GoodsReceipt process are going to be set up and drawn, which include the mock-ups and prototype of the screens, the conceptual workflow of process, architecture design of the application. By having and presenting the initial UI layout to the customers, the feedback is going to be collected from them in order to meet their necessities. Furthermore, these diagrams bring huge advantaged impacts on the implementation and testing process by avoiding missing important functionalities and developing redundant, useless parts; reducing refactoring and debugging time.

Figure. 18 The mock-up version of MDE before having new processes

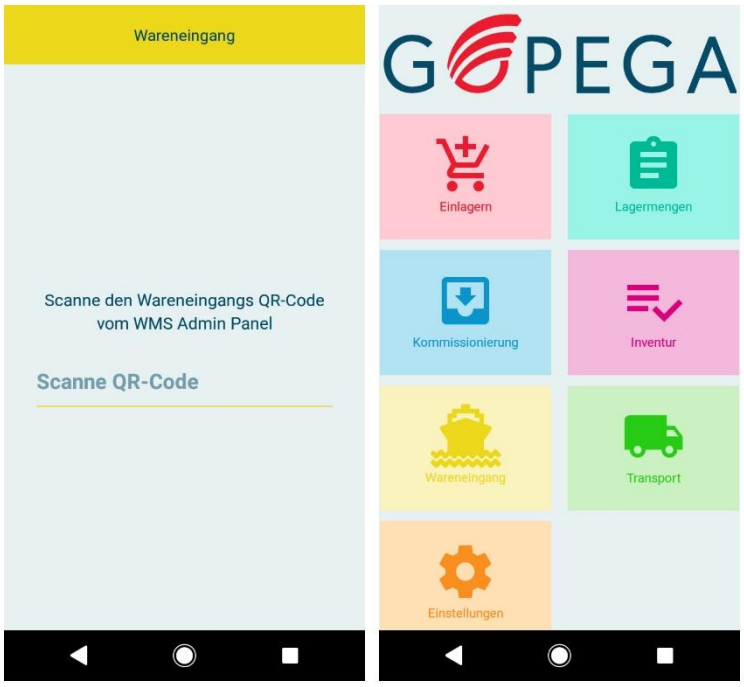

Figure. 19 The mock-up version of the MDE when having new processes including GoodsReceipt

Figure 20 displays the procedure of the GoodsReceipt and the other relevant processes in the entire system. The warehouse management system begins with the ordering actions from a customer in a web shop. After receiving the order, this web shop system forward it

to the middleware virtual cloud in order to convert any kind of data form into the WMS data schema, and to automatically generate the avis, or advice for this order. These documents are being sent to the WMS database and stored in the appropriate tables. By this way, the warehouse admins and retrieve any order or advice in order to create the goods receipt and save it into the database.

Certain steps take place in the warehouse storage and are done by the warehouse workers using the mobile application, in this case is the MDE. These users can scan the receipts QR codes and fetch the articles information of the corresponding advice, view them in the device screen. With the digital list in hand, the workers come to the warehouse interface, where the incoming products arrive, and look for the appropriate items to add into the goods receipt. This operation will update the information of the receipts and report any faulty outcome to the admin panel; therefore the managers can inform the customers about the status of their ordered products. In case the data of the articles from the advice and the entities information inside the goods receipt are matching to each other, the admins will accept them and create the transport process for moving the new articles to the warehouse internal storage.

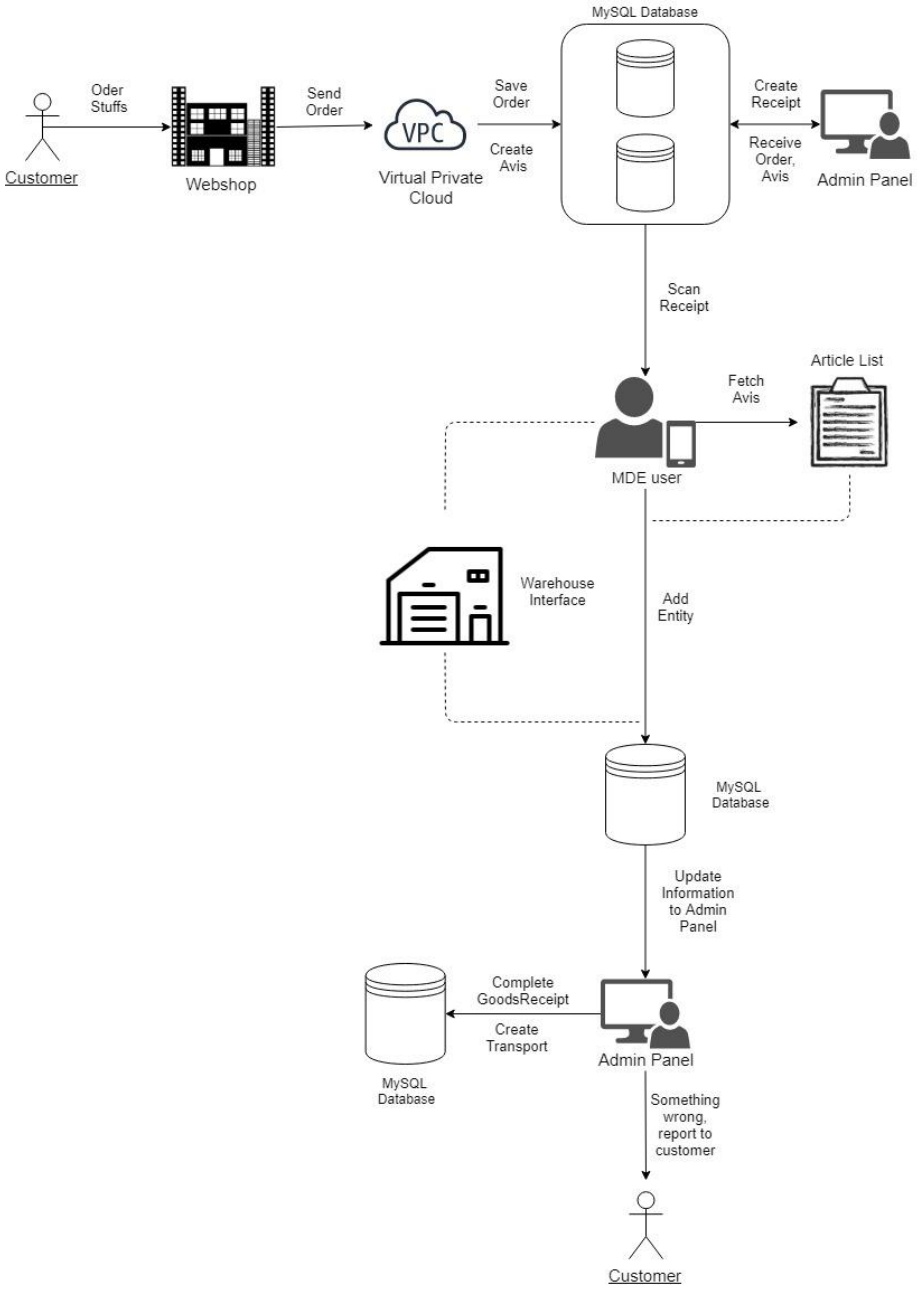

Figure. 20 GoodsReceipt and other related actions in the warehouse management system and external triggering actions.

5.3 Project initial setup

To get started with developing the application, it is important to choose and install an IDE and other needed packages. Visual Studio Code has been chosen for this case due to its lightweight running process and high integrated with other tools and plugins such as Gitlab, Github. In order to get and use this text editor, developer can easily download it from the Visual Studio main website: https://code.visualstudio.com.

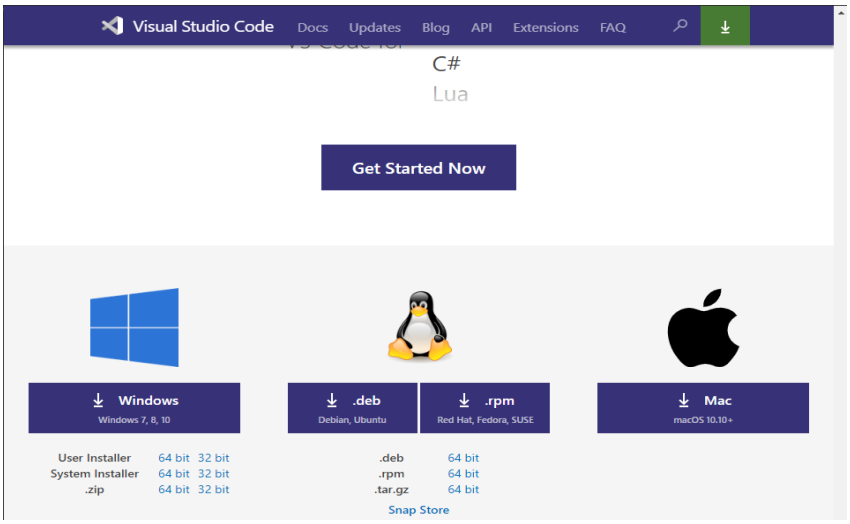

Figure.21 Homepage of Visual Studio Code

The next step in the setting up process is installing NPM software and React Native. This Node Package Manager will provide a large number of free packages which are able to be downloaded and used.

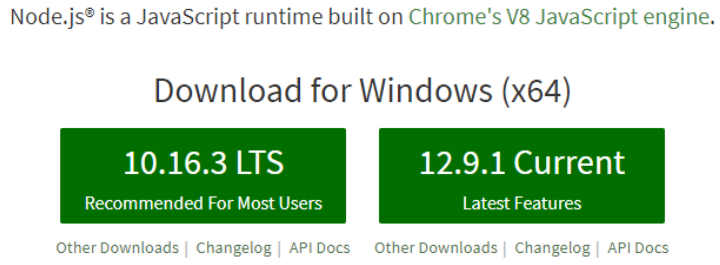

Figure. 22 Users can choose between different versions of NPM to download in Node.js Homepage

Once the Node packages is completely setting up, React Native and the other Javascript libraries can be installed via command line.

```
npm install -g expo-cli
```

Listing. 1 Command line for installing expo cli

```
npm install --save react-redux
```

Listing. 2 Command line for installing package for using redux in react native

```
npm install native-base --save
```

Listing.3 Command line for installing NativeBase library

The above command shows the installing method for the special library, named NativeBase, for developing cross-platform UI elements. It inherits the existing components API from React Native and combining with native theme, also provides a number of custom ones.

```
npm i react-i18next@legacy
```

Listing.4 Command line for installing i18n translation package

Beside of using NPM, React Native developer can use another packages manager for this application called Yarn. Developer can choose a specific version of Yarn which is match with other libraries. This software can be downloaded directly from the main website "https://yarnpkg.com/en/docs/install#windows-stable", or via other tools such as Chocolatey or Scoop. Afterwards, developers are able to download any package using the command: *yarn add package-name*.

5.4 Project implementation

After finishing installed all necessary tools and libraries, the implementation of the project can be started by forking the current version of the application via company frontend team

repository in Gitlab. The starting point of this system is the entry view with the login form as shown in the figure 18 above. Continuing from this screen, the tile menu will be shown with five buttons to enter the other processes inside the MDE. This is the place where GoodsReceipt part is going to be instantiated with the new entry button.

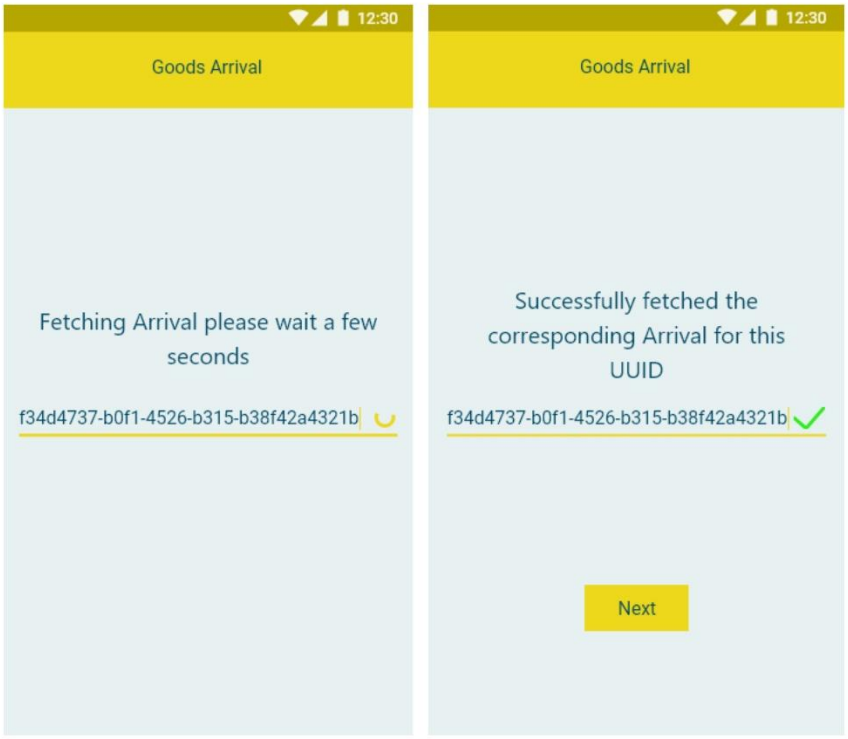

Figure. 23 Scanning QR code of the receipt.

The first screen to enter the process is scanning the QR code of the receipt. As mentioned in chapter 6.1.1, it can be filled in the input field which is centered in the content of the screen, while the header presents the title of the process. Above the input is the <Text> component containing the dynamical warning sign. Depending on each different state during the scanning goods receipt, it is rendered inside the conditional operator. If the input is empty, the instruction will be shown advising user to scan the code. During the fetching receipt, a spinner will appear besides the input, along with the different text, thus a green confirm icon or a red close circle icon will replace the spinner, if the QR code of the receipt is scanned correctly or not.

```
{!input ? (
  <Text style={styles.scannText}>{`${t('receipts:startscan')}`}</Text>
)
  : (isFetchingGoodReceipt ? (<Text style={styles.scannText}>{`${t('receipts:fetchscan')}`}</Text>)
    : (fetchingGoodReceiptError === undefined
      ? (<Text style={styles.scannText}>{`${t('receipts:error:scan')}`}</Text>)
      : (<Text style={styles.scannText}>{`${t('receipts:endscan')}`}</Text>)))}
```

Figure. 24 Using the conditional (ternary) operator instead of normal if-else to render the different content.

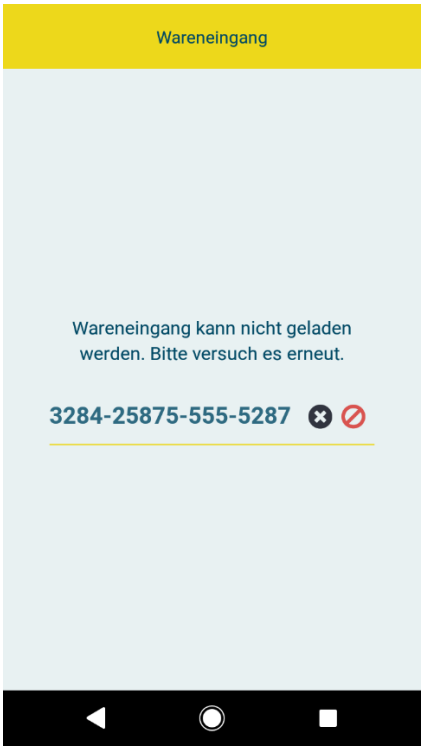

Figure. 25 Behavior of the screen when scanning the wrong QR code

These two figures below display the list of items which are fetched from the corresponding advice (figure.27) or from the internal resources (figure. 26). In both cases, the items are going to be demonstrated by using <FlatList> component in React Native library, which gets the different values for the data props. All the items in the list will be rendered indirectly through the renderItem props, in which a separate functional component is declared returning all of the JSX elements presenting the data of every item in the list. In the case there is no data passed to list, it is going to render a functional component showing a warning text via the ListEmptyComponent props. For instance, figure 26 shows the empty list with the gray paragraph advising user to search for the articles in the internal resources whether there is no corresponding advice relating to the receipt. Above

the list is the header of the screen containing an input field which allows user to search for the specific articles without scrolling the long list of items.

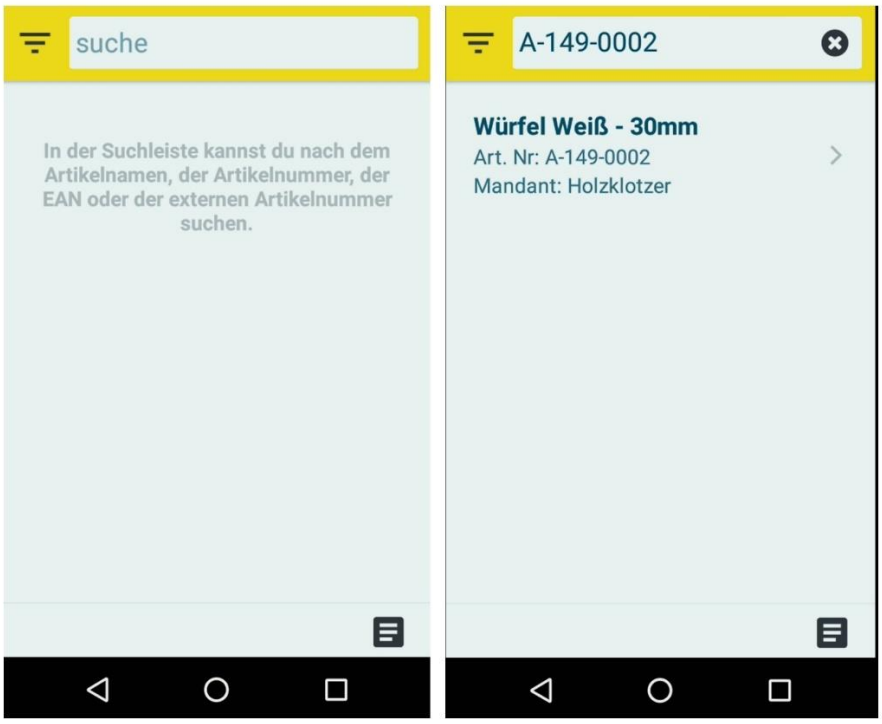

Figure. 26 List Item screen with no corresponding advice, has to show articles from internal resources.

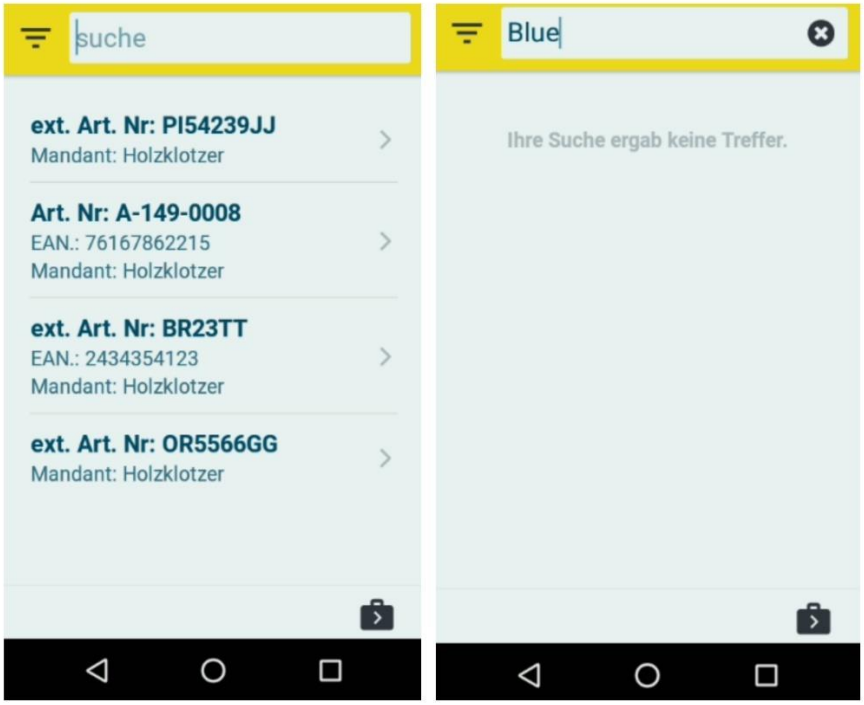

Figure. 27 Showing list of products fetching from the corresponding advices (left) and the behavior of the screen when search result is empty list (right).

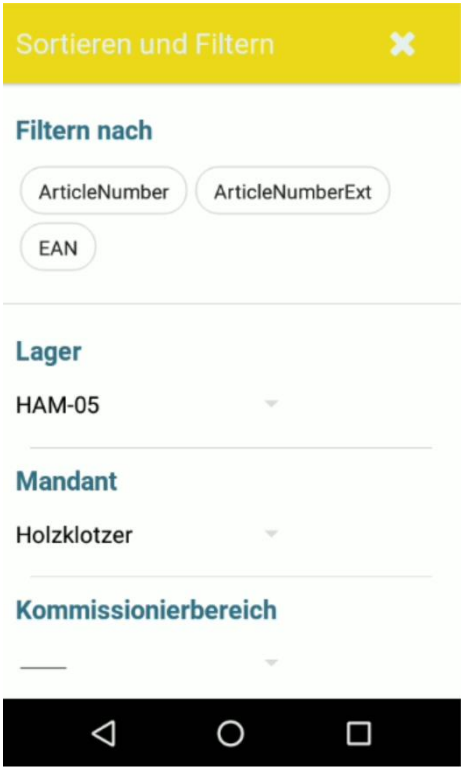

Figure. 28 Filter component for setting the global filter.

As shown in figure 28, a modal is rendered inside the button component in the header field of the screen. The visibility of this modal will be modified by changing its own visible props value when the user clicks on the button. By using this component, warehouse workers can update the global filter for the searching actions not only in the GoodsReceipt process but also the entire application. Articles can be found based on their different identical number such as article number, or an external article number. One distinct feature that this process does not have is changing the warehouse, client and pick zone filter due to the un-editable dropdown components. The idea behind this limited feature is that these values will automatically be updated by fetching the information inside the receipt after successfully scanning its QR code.

After choosing an article to be added to the goods receipt, the application will proceed to the next screens. There are two separate components needed to render depending on the information of the chosen item. If this article a number, the list screen will navigate to “SelectSerialNumber” screen, otherwise it will move to “SelectQuantity” screen. The only difference between these two is the first component to be rendered.

The “SelectQuantiy” screen is displayed in the below figure. There are two compulsory input fields inside the parent component, which are rendered in two independent child components. These components are going in to be rendered in the screen by importing them and passing the necessary props values without combining all the JSX elements in one file. After filling in an input field, the continue button is enabled for the user to move to the next step. Otherwise, this button is disabled.

The hardware back button navigates the user the to the last screen in the navigation stack. There is another back button in the footer of the screen, which is above the device navigation bar. Instead of having the same function as the hardware back button, it allows user to return to the last step in the scanning process, which in this case is removing the container input and re-rendering the quantity input. This approach is done via conditional rendering when changing the state of the value of the quantity input to null.

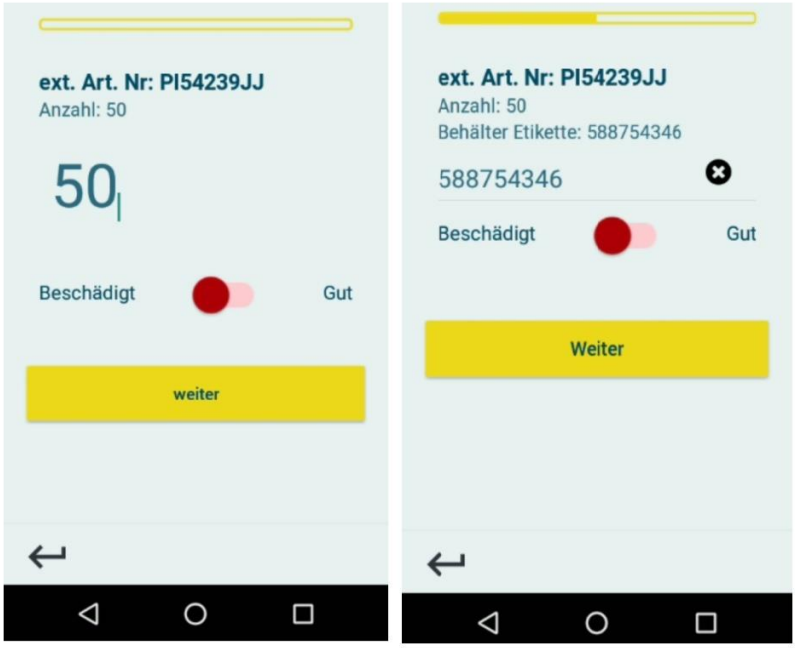

Figure. 29 Scanning the quantity, the container and the condition of the incoming article inside a screen.

The last screen of the process is the overview showing all the information which can be retrieved from the filled input in the previous screen and the data inside the advice. The yellow button in the figure. triggers the action of adding new entity to the goods receipt. Users still have the opportunity to undo their tasks by clicking the close button to navigate to the list items screen or pressing the hardware back button to go back to the previous scanning input screen.

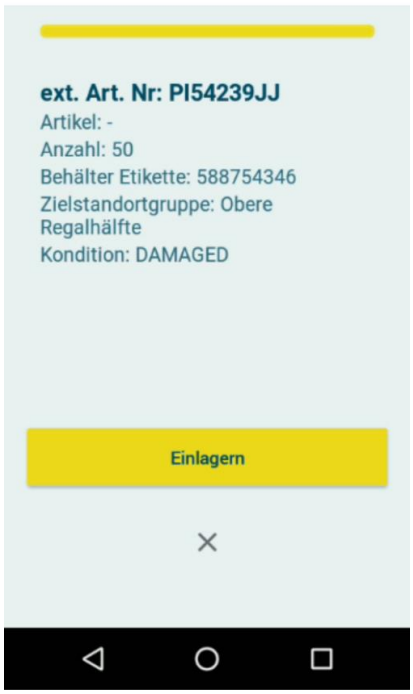

Figure. 30 Overview screen showing the filled information.

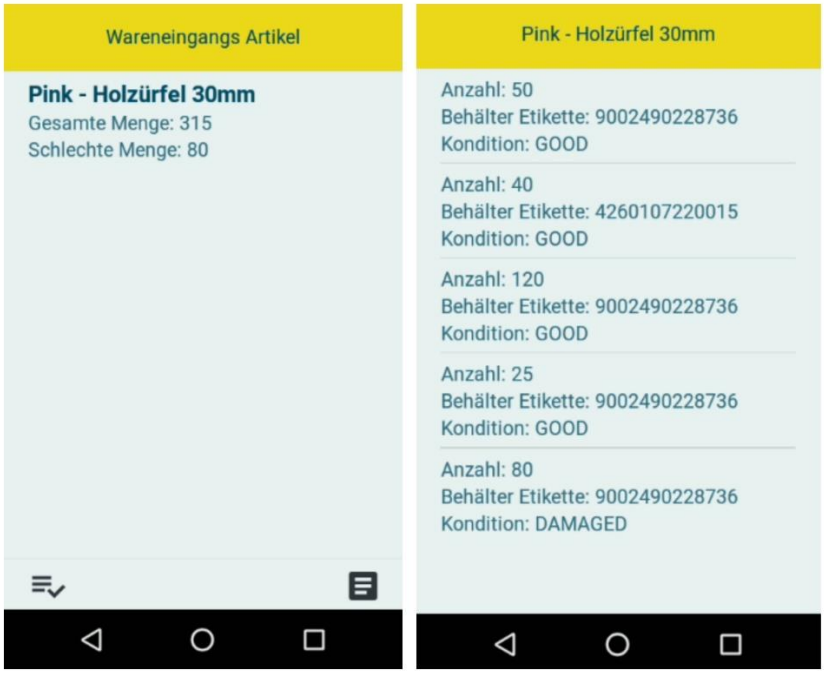

Figure. 31 List articles which is added via

After finishing the add entity process in the GoodsReceipt, warehouse workers are allowed to review their work, by reviewing the list of added items. In the bottom right of the list screen, there is a button icon navigating to the list articles which are just added to the receipt. Clicking on the specific article will move to the detailed screen displaying all the entities information.

5.5 Project discussion and summary

The release of the new version of MDE application has gained plenty of positive feedback from warehouse workers. By reducing the amount of unnecessary displayed information and using meaningful customized components, the GoodsReceipt feature has been processed effortlessly with the limitation of the errors. By utilizing the React Navigation, all the steps inside this procedure are handled and connected together in a flexible way, which help the users to perform various tasks faster.

The integration between Redux and the middleware plugins has been improving the application performance significantly. All the relevant data fetched from the server are being modified and re-created inside the reducers, instead of being manipulated directly in the components. As a consequence, these states are not updated automatically in the entire component lifecycle, especially during the rendering progress which can lead to slower compiler time and runtime of the application.

With the new database structure and the requests from the customers, several changes will be updated in the near future for the better process between the warehouse systems and their own web shop clients. The design of the application logo will be also renewed while the color theme of the entire application is still unchanged. These modifications will become the requirements for the next major release of the MDE, which are going to be launched in parallel with the new admin panel desktop application. By using expo and yarn as the package manager, every update in the mobile application will be applied directly into the current version. Therefore, users just receive a notification about the updates without removing the old one and re-installing the entire application.

6. Conclusion

This thesis demonstrates the usage of React Native in warehouse management systems, especially in developing cross-platform mobile application for a warehouse procedure. Warehouse area and logistics have been significant parts of the industry for decades. In order to meet the need of the market, applying new technologies will improve the performance and cost efficiency of warehouse and management systems. Along with other modern programming languages and tools such as Java, Javascript frameworks Angular Js, Vue Js, the debut of React Native have brought a powerful option for the development of mobile application.

For this study, several relevant topics with the concrete examples have been illustrated. It demonstrates the fundamental philosophy of the warehouse management and logistics industry so that the reader can understand the entire process of these fields. Furthermore, certain warehouse applications are presented along with their advantages and drawbacks throughout the evolution of the warehouse management systems. As a result, the requirement of React Native is inevitable for the improvement of the warehouse systems.

React Native is a new framework which was developed and released by Facebook in March, 2015. This Javascript library has lots of advantages which affect the developer community and software industry, especially in mobile development, compared with other similar technologies. There are light-weight tools which can integrate with other libraries and external tools. With using the modular concept, React Native code is easy to maintain and furthermore, it is reusable. Also, the performance of React Native in implementing mobile application in both platforms Android and iOS is as good as the native development. Consequently, this advanced technology has been chosen for designing a mobile scanning application, especially in the GoodsReceipt process, for a warehouse in Hamburg, Germany.

The launching of MDE, the scanner mobile application, has been part of the new software solutions for warehouse management especially with the new feature of GoodsReceipt, It allows workers to boost their performance in the process of receiving and verifying new products before allowing them to be stored in the internal warehouse facilities. After the first release of the new feature, it has gained positive feedback from the customers and has already been used in the production phase.

Despite of the convenience, developers are still facing a number of struggles when using React Native in their application. The newest released version of React Native, which is numbered as 0.60, has been considered as the most stable version. Even though, this new Javascript framework is still an immaturity technology which can bring negative impacts on the applications. However, with those advantages in both iOS and Android devices, React Native is a promising technology to be used in mobile and logistics industry. With the continuous contribution from Facebook and the developer community, this framework has a profitable and bright future as being the top considered option for the application development.

References

1. Stack Overflow developer survey results in 2017 [online].

URL: https://insights.stackoverflow.com/survey/2017#technology

2. Tech MeetUps article [online]. Most popular and influential programming languages in 2018.

URL: https://techmeetups.com/most-popular-and-influential-programming-languages-of-2018/

3. Zion Market Research Citing references [online]. Warehouse Management Systems Market Fostering to Develop and Reach USD 3.04 Billion By 2022.

URL: https://www.zionmarketresearch.com/news/warehouse-management-systems-market

4. Michael ten Hompel, Thorsten Schmidt. Warehouse Management: Automation and Organization of Warehouse and Order Picking Systems; Chapter 2; 2007.

5. IBM Icon of Progress Citing references [online]. Information Management System.

URL: https://www.ibm.com/ibm/history/ibm100/us/en/icons/ibmims/

6. Adam Robinson. The Evolution and History of Supply Chain Management [online].

URL: https://cerasis.com/history-of-supply-chain-management/

7. Thomas Lum, Dick K. Nanto Citing references [online]. CRS Report for Congress. China's Trade with the United States and the World.

URL: https://fas.org/sgp/crs/row/RL31403.pdf

8. Enlyft article [online]. Companies using SAP Extended Warehouse Management.

URL: https://enlyft.com/tech/products/sap-extended-warehouse-management

9. Vietnamnet report [online]. Top 500 largest enterprises in Vietnam announced; December 11th, 2018.

URL: https://english.vietnamnet.vn/fms/business/214330/top-500-largest-enterprises-in-vietnam-announced.html

10. Nicole Pontius in Industry Resources [online]. Top Mobile Apps for Warehouse Managers: 50 Useful Apps for Managing Inventory, Shipping and Tracking, Workflow and More.

URL: https://www.camcode.com/asset-tags/top-mobile-apps-for-warehouse-managers/

11. Bonnie Eisenman. Learning React Native [online]; O' Reilly Media, Inc; December, 2015.

URL: https://learning.oreilly.com/library/view/learning-react-native/9781491929049/ch02.html#introduction-concepts

12. React Documentation: React Component [online].

URL: https://reactjs.org/docs/react-component.html

13. React Native: Layout with Flexbox [online].

URL: https://facebook.github.io/react-native/docs/flexbox

14. Eric Masiello, Jacob Friedmann. Mastering React Native [online]; Packt Publishing; 2017.

URL: https://learning.oreilly.com/library/view/mastering-react-native/9781785885785/ch05s02.html

15. John A. Calderaio. Comparing the Performance between Native iOS (Swift) and React Native; February 22nd, 2017.

URL: https://medium.com/the-react-native-log/comparing-the-performance-between-native-ios-swift-and-react-native-7b5490d363e2

16. Jakob Nielsen. 10 Usability Heuristics for User Interface Design; April 24th, 1994.

URL: https://www.nngroup.com/articles/ten-usability-heuristics/