

The online fashion
operation platform

by

Chan Chi Man

Submitted in partial fulfillment of the requirements for the
degree of

Bachelor of Science (Honours)
in Computer Science

Hong Kong Baptist University

April, 2020

Declaration

I hereby declare that all the work done in this Final Year Project is of my independent effort. I also certify that I have never submitted the idea and product of this Final Year Project for academic or employment credits.

Chan Chi Man

Date: _____

Hong Kong Baptist University
Computer Science Department

We hereby recommend that the Final Year Project submitted by Chan Chi Man entitled “The online fashion operation platform” be accepted in partial fulfillment of the requirements for the degree of Bachelor of Science (Honours) in Computer Science.

Dr. CHOY, Martin Man Ting
Supervisor

Dr. FENG, Jian
Observer

Date: _____

Date: _____

Abstract

As a cosmopolitan city, Hong Kong is also a fashion city. Hong Kong has trained a lot of young and creative professional fashion talents. Some of them can step out of Hong Kong to enter the international arena and promote the integration of Chinese and Western cultures and Hong Kong's unique culture to the world. However, at present, there are not many ways for new designers to promote their designs to other places. The purpose of this project is to develop and provide a cross-platform web platform for young designers to promote their finished designs.

Table of Contents

Abstract	i
1 Introduction	1
2 Proposed Solution	2
2.1 Scope of the proposed solution	2
2.2 Architecture of the proposed system	3
2.3 Role of Users	3
3 Research	4
3.1 Mobile App Platform	4
3.1.1 Native App [2]	4
3.1.2 Hybrid App / Cross-platform App [2]	5
3.1.2.1 Xamarin	6
3.1.2.2 React Native	7
3.1.2.3 Flutter	9
3.1.3 Web App / Progressive Web Application [2]	10
3.2 Ecommerce Platform	11
3.2.1 Magento	11
4 Design of the solution	12
4.1 Frontend	13
4.2 Database	15
4.2.1 ORM	15
4.2.2 EAV	16
4.3 API	18
4.4 React Native	21
4.4.1 Lifecycle	21

4.5	Image Classification	24
4.6	UI Design - Mobile Platform	25
4.6.1	Connected with Magento API	25
4.6.2	Withoutconnect Magento API	25
4.7	UI Design - Web Platform	25
5	Critical Evaluation	33
5.1	Changes to design and justifications	33
5.2	Quality of implementation	34
5.3	Development Difficulties	34
6	Result and conclusion	35
6.1	Future development	35
Appendix 1: Project Schedule		37
References		38

Chapter 1

Introduction

An online store that provides backend management systems gives emerging designers and smaller brands a platform for them to place their designs. The difference between this platform and other platforms is that designers and brands can cooperate with each other. There can be finished products of different designers in a piece of clothing advice provided to consumers, and it is no longer limited to the same brand or designer's products. At the same time, designers no longer need the platform as an intermediary to communicate with customers. They can directly contact customers to understand their needs and provide corresponding solutions, reducing the time required for intermediary communication and speeding up their problem resolution.

Cross-platform mobile applications not only expand designer and brand promotion channels, but also allow them to monitor and manage their product data at any time and place. As mobile shopping becomes the daily life of the general public, a cross-platform mobile application is necessary and an important way to give consumers know more new brands and promote them to more people through their social platforms.

Chapter 2

Proposed Solution

2.1 Scope of the proposed solution

The solution should include the following components.

An online store with backend system

- Provides data analysis to advise designers on brand marketing
- Connect to different social networks, and buyers and sellers can communicate using social platforms
- Designers can edit and monitor their brand 7 * 24 hours
- Provides the gateway between connecting with web system and mobile system
- Processing image recognition for mobile platform

Cross-platform mobile application

- Designers can edit and monitor their brand 7 * 24 hours
- Allows users to find similar styles of clothing on our platform using a smartphone camera
- Allows users to do the searching with speech recognition
- Offline recommend products view

2.2 Architecture of the proposed system

- Software
 - Magento 2
 - MariaDB
 - React Native
 - Tensorflow
- Hardware
 - Android and iOS system smart phone
- Internet Access
 - Wi-Fi
 - 4G LTE

2.3 Role of Users

- Designers / small brands
 - They can built up and promote their shop easily.
 - Control everything just like they has own a website.
 - They can control and analyse the data to help make the business decisions.
- Customers
 - Buy different products from local designer and get the matching suggestion from designer.
 - They can contact the designer directly if there are any questions.

Chapter 3

Research

3.1 Mobile App Platform

Since the opening of Apple App Store in 2008, the number of mobile applications has reached more than 2 million on both Android and iOS platforms in more than a decade. [1] Today, urbanites use their mobile phones to access the Internet every fat to do a variety of things, including online financial management, Instagram and trading. Some people can forget to take their wallets out of the street, but they can't go out without their smart phones.

The development of mobile applications is also all the rage. To this day, mobile application development has gradually evolved into different types, and companies can choose the type of development suitable for the enterprise's mobile application according to their needs, resources, development time, and so on.

3.1.1 NATIVE APP [2]

Native app means develop an application using the native programming language of the mobile system. From the earliest ObjectiveC to Swift for iOS, or Java to Kotlin for Android, are the most direct development methods. In theory, native applications should be able to make the most of all the

capabilities of the phone and achieve the best user experience.

Native applications can change immediately following the upgrade of the mobile phone system, so it can make fuller use of the functions of the phone than other development types of applications. For example, machine learning, or Augmented Reality, epoch has been quite popular in recent years, is a native application that can be used for the first time.

In contrast, the disadvantage of native applications is that the development cost is relatively high. In the iOS or Android market, sometimes even PC users need to be taken into account. Unless under special requirements, it is basically necessary to support users on more than two platforms at the same time. With multiple platforms being developed separately, it is not uncommon to double the cost and time of development.

3.1.2 HYBRID APP / CROSS-PLATFORM APP [2]

When the problems of time and development cost of native applications become more and more obvious, the market naturally has relative solutions at the historic moment, and this is the hybrid application.

The basic concept of a hybrid application is to place a web browser in the smart phone. Through this browser, you can open a website that simulates a mobile phone application. Because browsers are common to mobile phones, all programs that can be shared are presented in the form of web pages. For the functions that the web page fails to implement, it will implement by native programming language. Taking the application of Android and iOS platforms as an example, a hybrid application can achieve twice the development cost and time of a native application by about 1.5 times the development cost and time.

In 2015, Ionic, which used AngularJS and Cordova as its foundation, was a madman, and a large number of applications written in Ionic emerged. However, the core problem of the hybrid application at that time was the performance of the user interface performance. At the time, the browser performance of Android and iOS was very limited. In terms of user experience,

hybrid applications were much worse than native applications.

In order to reduce development costs without giving up the user experience, various mobile application companies continue to explore, of which Cross-platform App is the successor of hybrid applications. The biggest difference from hybrid applications is that they improve the user interface experience. Instead of using web pages and browsers, they provide a common user interface development framework for developers to develop native user interfaces. This greatly improves the user experience of hybrid applications. Facebook's React Native, Microsoft's Xamarin, Google's Flutter or Vue Native, these are cross-platform application development frameworks.

However, although these development frameworks all claim to be development tools for native applications, they are quite similar to native applications in terms of user experience and performance. But cross-platform applications often take at least a few months to support the latest smart phone features, which is always not as fast as native applications. Even so, cross-platform applications can still meet the needs of more than 90% of users. Mobile applications that you often come across, including Instagram, Evernote, UBER, Twitter, Netflix, etc., are all cross-platform applications, which shows that cross-platform applications have become the mainstream of today's development model.

3.1.2.1 Xamarin

Xamarin was born in 2011. The first set of products was Xamarin.Mac in 2012. iOS apps can be developed using C# language and successfully listed on Apple's App Store. Xamarin was bought and integrated by Microsoft shortly thereafter. In Visual Studio, the use of Xamarin to develop App in the past required additional payment, but now as long as you download Visual Studio Community, you can directly develop Xamarin related applications.

The main feature of Xamarin is that developers can directly use C# to call APIs of Android and iOS to generate a native interface. The architecture is basically very similar to the native language architecture (Android, iOS).

The development is almost the same. The advantage of using Xamarin to develop cross-platform apps is that the general logic program can be used directly between the two projects of Xamarin.Android or Xamarin.iOS. But we still have to call the respective underlying API if it need to render the UI part. In other words, Xamarin only unified the development language, in fact, you must still understand some Android and iOS development frameworks to develop.

From the perspective of Github's community thermal network, although Xamarin has launched products as early as 2011, the thermal network between developers seems to be not high, because Xamarin uses the underlying API to present the UI, so the screen presentation between different devices can sometimes be very different, for example, TabbedPage is very different on different devices can sometimes be very different, for example, TabbedPage is very different on Android and iOS.

In addition, Xamarin's Hot Reload can only be used to render XAML UI. Although it supports fast deployment, it still seems to require a long compiling time.

Based on the above conclusions, developers who are more suitable to use Xamarin as a development tool should be:

1. Have C# development experience
2. Developers accustomed to the Visual Studio development language as the main development tool
3. Developers who have a certain understanding of the Android or iOS UI layer framework
4. Newcomers who want to quickly develop a simple business process App

3.1.2.2 React Native

React Native is a cross-platform development framework that Facebook opened its source code in May 2014. The development language is Javascript. The syntax and architecture are very similar to the front-end web development React.JS JSX and CSS. It allows front-end web developers to quickly

enter the development area of React Native App.

The framework of the React Native operation is built with Javascript. Javascript can directly call Native to build the UI. In addition, React Native also supports real-time hot reload. In other words, React Native does not need to be re-run during the Run-time stage. Compiling programs can directly control the control operations of the UI or program logic, and the execution performance is also very close to the Native development performance.

Observed by the popularity of the community, React Native can be regarded as a very fast and popular development framework in recent years. The main reason should be attributed to the success of the React.JS development model. Many React.JS front-end web developers can quickly enter the ranks of mobile device application development. In addition, Facebook and Instagram apps are also developed using React Native. I believe that many players who are curious about the function of Facebook to update components locally should want to understand how Native React it works, but React Native uses Bridge to call the underlying Native UI through Javascript after all, so the consistency of the UI appearance of different operating systems depends on the third-party components.

However, taking Tabbled Page as an example, compared to Xamarin.Form, React Native's Tanned Page has made it possible for the display of operating system screens of different platforms to be quite similar.

Relatively speaking, the React Native framework is very suitable for developers using these conditions:

1. Those who only understand Web front-end development technology, because they only need to understand Javascript and CSS to understand React Native code.
2. Developers who like React.JS development architecture
3. Developers who need to develop cross-platform apps quickly

3.1.2.3 Flutter

Flutter was announced by Google at the 2015 Flutter conference. The official stable version was announced on December 4, 2018. The bottom layer mainly uses C++ for development to connect iOS and Android, and uses Google's Skia graphics library to provide the bottom graphic support. All of Flutter's UI is composed by combining various Widget. The basic library is written by Dart, which provides the categories and libraries required by Flutter.

Flutter's hot reload can inject the modifications from original file into the running application. Flutter expands this function by supporting stateful hot reload. In most cases, changes to the source code can be immediately reflected in the executed application without restarting or losing any state.

Different from the way that Xamarin and React Native call the native interface, the biggest difference of Flutter is that the interface of Flutter is derived through the 2D engine Skia Render, so the performance is even better than the native application. Because all the UI is rendered, the screen output by Flutter are almost the same on iOS and Android.

Judging from the popularity of the community, Flutter has grown faster than React Native. According to the number of followers on Github, it has already surpassed React Native. It should be expected to become the most popular cross-platform development tool.

However, compared with Xamarin and React Native, the stable version of Flutter has only been released to the present in the past year or so. The stability of the framework has yet to be tested by time. The method of direct rendering is likely to replace the original development framework that use Native UI, and become a new generation of popular cross-platform device development trend.

Flutter has a high consistency on the screens presented on different devices, and developers can reduce many maintenance costs arising from the inconsistency of output functions.

What types of developers is Flutter suitable for:

1. Requires consistency in screen display across platforms
2. The product needs to be online in the fastest time
3. Those with higher UI performance requirements

3.1.3 WEB APP / PROGRESSIVE WEB APPLICATION [2]

To put it plainly, the Web App is just a web page, but this kind of web page is mostly built using the framework of Single page Applications. With the development of browsers on mobile phones in recent years, web apps have been able to give users the feeling of being very close to native applications.

In terms of performance, the performance of web apps in current browsers has improved a lot compared to a few years ago, but of course it is inferior to the two mentioned above. In terms of functions, the web app also has many functions that cannot be achieved, such as reading QR Code, Bluetooth device information, etc., and cannot use paid functions such as Apple Store or Play Store. However, if the enterprise's application is mainly based on simple data such as text and pictures like a shopping network, the web app is definitely a good choice.

Based on the advantages of web apps, iOS and Android also made corresponding performance enhancements, so Progressive Web App appeared. When you use Safari on iOS or Chrome on Android to browse the web, users can choose to add the web page to the desktop of the phone, so that the web page will appear as a mobile application, for the average user, PWA looks no different from other mobile applications. And these applications can also use the functions that many native applications have, such as pushing information, reading coordinates, taking pictures with the camera, using the mobile phone compass, and so on.

Web app and Progressive Web Application also have a very prominent advantage, that is, it can be used directly without downloading through the App Store. There are many reasons why users refuse to download native apps, such as limited Internet data, insufficient phone capacity, and too many apps installed on their phones. The web app and PWA are to dispel the concerns of these users. Users can simply scan the QR Code or search on

Google to start using it. This fast experience is quite effective for promoting new products.

The most important thing is that the development cost and time of the web application is the shortest among various development models. With the lowest cost, it can support almost all mobile phone platforms, even computer desktops can be used simultaneously. It is very suitable for some such as online shopping, it is a service that does not require many functions of the mobile phone to reserve a room, etc..

3.2 Ecommerce Platform

3.2.1 MAGENTO

Magento is an open source e-commerce platform written in PHP programming language; it is mainly for enterprise-level applications and can handle various needs.

The technical requirements of Magento's website construction are relatively high, and the cost of operation and maintenance is relatively large, so Magento is suitable for enterprises with many types of SKUs, large traffic, and high investment. The large traffic is because some e-commerce systems like Opencart, the system performance will be reduced when the traffic is large, and Magento will be more stable.

Magento is designed to be very flexible, with a modular architecture system and rich functions, which is easy to integrate seamlessly with third-party applications. Since magento is open source, the code is completely in your own hands, so you can personalize all the content of the website according to your needs, including Shopping flow, page content, custom buyer show, reviews, etc. can think of everything. Thousands of special function plugins for Magento have been developed by global enthusiasts and enterprises, which can be turned on and off at any time after installation. In addition, Magento does not have any platform commissions. The server, source code, and database are owned by the business owner.

Chapter 4

Design of the solution

As mentioned above, the core problem that I want to solve would be the very good works from our fresh designer are buried and do not have chance to let others know. This problem is mainly caused by designers need to have a many connections before they can built up their own brands. However, the fresh designers may not have the chances or platforms to know more people and get resources to promote their well products to other places. If there is a platform that provide a chances to let them promote their products and learn more built up skills for their further usage.

By finding the suitable solution of this project, I have studied a similar cases done by Hong Kong Polytechnic University.

For the platform which is ITC Store and is done by Hong Kong Polytechnic University, it is found the their platform has some benefits and disadvantages that fulfil the need of this project.

The benefits:

- Connect with the offline store, the customer can be take a look of the product before they buy from web store
- Provide a online store for other counties have a way to know more our local designer products
- There has a middleman to monitor the uploaded information

The disadvantages:

- Designers don't have any control of their products on the website
- There do not provide any data analyse service to the designers
- Customers need to contact to the designer through the middleman, it takes time for the communication

It is found out the studied case neither solve the problems in Hong Kong, thus we need to come up with the following solutions to solve the problems.

As the problem of designers don't have any control of their products on the website, I would build up a web store with a backend system and mobile application for designers to upload the products and have different settings.

As the problem of lacking contact to the designer, customers can contact the designer through chat room from the web store. The web store and mobile app can show the newest product from the local designers. All the sales data will upload to the server and database for further analysis which provide to the designers.

After doing the research, the React Native in cross-platform application is the most suitable for this project to complete in a short period.

4.1 Frontend

Magento 2 is not a Model, View, Controller system. It is a system similar to Model, View, ViewModel (MVVM), but the architects have not given it a specific name.

In Magento 2, when you request a URL, the system will route the request to execute method of the Controller class, similar to the MVC system that routes the request to the action method of the Controller class. However, unlike the traditional MVC system, this Controller class is only responsible for the following things:

- Decide which page layout to use

- Handling saving data from POST requests
- One of the following two things:
 - Let the system render HTTP response
 - Redirect the user to the next or previous page

Each View is responsible for obtaining its own information from the model layer, request object, or other external systems. Magento decomposes the HTML page into many fragments, called containers. Each container contains a nested tree of objects, called blocks. Each block object has a phtml template file, which is used to specify the HTML to be displayed by the block object. According to MVVM, the block object of Magento is the View Model. The block object will complete all data reading into CRUD models, request object, external systems, etc. The phtml template file is View in MVVM, it only interacts with ViewModel.

The traditional PHP MVC system sets the variable values in the view in the controller. Magento's approach is a bit different from the traditional one. Magento 2 uses Model, View and View Model to separate business logic and template logic. For a large development team, everyone's responsibilities are clear. This change of Magento is likely to be more beneficial to them, but it is not very good for full-stack engineers. Full-stack engineers need to consider more levels of abstraction.

Like most modern frameworks, Magento 2 uses many different cache files to speed up operations that will slow down. The purpose of these caches is to make the system run faster in production environment, but this often results in Magento not using the latest configuration or source files, so no changes can be seen.

When create new features in the Magento 2 system, it is often necessary to clear the cache. It can use the CLI command of bin/magento, System -> Cache Management in the backend, or delete the cache file to clear Cache.

In order to speed up the system, in addition to the cache file, Magento 2 also generates many boilerplate classes. These files are placed in the var/-generation path. When there has modify some configuration or code files, it is usually necessary to regenerate these files. Currently, there is no method,

ever CLI or backend, that can help regenerate. The only way is to manually delete the files under var/generation.

4.2 Database

4.2.1 ORM

Magenta is a very complete packaging framework. In addition to practicing the spirit of many design patterns, there are also some other patterns in it, such as the ORM architecture is one of them.

ORM, whose full name is Object-Relational Mapping, is a programming pattern used to convert data between different types of systems in object-oriented languages. Its role is to make an encapsulation between the associated database and the entity, so that when we specifically operate Object, we do not need to deal with complex SQL statements, only simple operation of the object Property and Method.

During development, it often interacts with the database. Whether it is in the Controller, View or Model, it is possible to use the four major functions of the database: Insert, Delete, Select, Update and through ORM to help us package the method, we can quickly get the data we want or achieve the operation we want to do to the database.

In Magento 2 architecture, the principle of one data table is mainly corresponding to one Model. Collection is a type of operation database in Magento, which implements many database select methods, let us use a very intuitive way to obtain the data in the database, and do not need to use the long SQL syntax. Model in Magento can be said to be an entity, whether the method of insert, update and delete, can be operated through Entity. There are two ways to obtain a Collection in Magento 2 to operate the ORM. The first is to declare the Model Entity first, which is obtained by using the `getCollection()` method in the Entity, and the second is to directly create a Collection Class.

Using Magento to operate the database through the ORM method not only increases the readability and maintenance of the program, but also greatly improves the security of the program. It is a good way to prevent SQL injection attacks.

4.2.2 EAV

In the Magento database, there are more than three hundred data sheets, and the data structure used in it is EAV. EAV is a relationship model derived from database management technology, where E of EAV is an Entity, each entity represents an object in the Magento system, and is stored in the database with an independent ID, such as, products, orders, customers, catalog, etc. A represents the Attribute, the attribute is the nature of each object in the Magento system, for example, the product has a name, price, quantity, etc. And V is the Value, which is the value of the attributes. Magento is through such a data model to access data in various formats.

Entity is defined as entity type in Magento internal program, and the built-in Entity type has the following eight types, which can also be added manually, and the new Entity type will be added in the table of `eav_entity_type`.

	<code>entity_type_id</code>	<code>entity_type_code</code>	<code>entity_model</code>	<code>attribute_model</code>	<code>entity_table</code>	<code>value_table_prefix</code>	<code>entity_id_field</code>	<code>is_data_shar</code>
1	customer	<code>Magento\Customer\Model\ResourceModel\Customer</code>	<code>Magento\Customer\Model\Attribute</code>	<code>customer_entity</code>	<code>customer</code>	<code>customer_</code>	<code>customer_id</code>	<code>null</code>
2	customer_address	<code>Magento\Customer\Model\ResourceModel\Address</code>	<code>Magento\Customer\Model\Attribute</code>	<code>customer_address_entity</code>	<code>customer_address</code>	<code>customer_address_</code>	<code>customer_address_id</code>	<code>null</code>
3	catalog_category	<code>Magento\Catalog\Model\ResourceModel\Category</code>	<code>Magento\Catalog\Model\ResourceModel\Eav\Attribute</code>	<code>catalog_category_entity</code>	<code>catalog_category</code>	<code>catalog_category_</code>	<code>category_id</code>	<code>null</code>
4	catalog_product	<code>Magento\Catalog\Model\ResourceModel\Product</code>	<code>Magento\Catalog\Model\ResourceModel\Eav\Attribute</code>	<code>catalog_product_entity</code>	<code>catalog_product</code>	<code>catalog_product_</code>	<code>product_id</code>	<code>null</code>
5	order	<code>Magento\Sales\Model\ResourceModel\Order</code>	<code>Magento\Sales\Model\Attribute</code>	<code>sales_order</code>	<code>sales_order</code>	<code>sales_order_</code>	<code>order_id</code>	<code>null</code>
6	invoice	<code>Magento\Sales\Model\ResourceModel\Invoice</code>	<code>Magento\Sales\Model\Attribute</code>	<code>sales_invoice</code>	<code>sales_invoice</code>	<code>sales_invoice_</code>	<code>invoice_id</code>	<code>null</code>
7	creditmemo	<code>Magento\Sales\Model\ResourceModel\Creditmemo</code>	<code>Magento\Sales\Model\Attribute</code>	<code>sales_creditmemo</code>	<code>sales_creditmemo</code>	<code>sales_creditmemo_</code>	<code>creditmemo_id</code>	<code>null</code>
8	shipment	<code>Magento\Sales\Model\ResourceModel\Shipment</code>	<code>Magento\Sales\Model\Attribute</code>	<code>sales_shipment</code>	<code>sales_shipment</code>	<code>sales_shipment_</code>	<code>shipment_id</code>	<code>null</code>

As mentioned above, each entity has different attributes, so each Attribute will correspond to the Entity type, such as the product name, price, quantity, etc., or the customer's name, phone, address, etc., and all Attributes will be stored in eav_attribute in this table, for customers with entity_type_id of 1, the blocks in the eav_attribute table are as follows.

attribute_id	entity_type_id	attribute_code	attribute_model	backend_model	backend_type	backend_table	front
1	2	1_website_id	<null>	Magento\Customer\Model\Customer\Attribute\Backend\Website	static	<null>	<null>
2	2	1_store_id	<null>	Magento\Customer\Model\Customer\Attribute\Backend\Store	static	<null>	<null>
3	3	1_created_in	<null>	<null>	static	<null>	<null>
4	4	1_prefix	<null>	<null>	static	<null>	<null>
5	5	1_firstname	<null>	<null>	static	<null>	<null>
6	6	1_lastname	<null>	<null>	static	<null>	<null>
7	7	1_lastname	<null>	<null>	static	<null>	<null>
8	8	1_suffix	<null>	<null>	static	<null>	<null>
9	9	1_email	<null>	<null>	static	<null>	<null>
10	10	1_group_id	<null>	<null>	static	<null>	<null>
11	11	1_dob	<null>	Magento\Eav\Model\Entity\Attribute\Backend\Datetime	static	<null>	Magento'
12	12	1_password_hash	<null>	Magento\Customer\Model\Customer\Attribute\Backend>Password	static	<null>	<null>
13	13	1_pc_token	<null>	<null>	static	<null>	<null>
14	14	1_pc_token_created_at	<null>	<null>	static	<null>	<null>
15	15	1_default_billing	<null>	Magento\Customer\Model\Customer\Attribute\Backend\Billing	static	<null>	<null>
16	16	1_default_shipping	<null>	Magento\Customer\Model\Customer\Attribute\Backend\Shipping	static	<null>	<null>
17	17	1_taxvat	<null>	<null>	static	<null>	<null>
18	18	1_configuration	<null>	<null>	static	<null>	<null>
19	19	1_created_at	<null>	<null>	static	<null>	<null>
20	20	1_gender	<null>	<null>	static	<null>	<null>
21	21	2_disable_auto_group_change	<null>	<null>	static	<null>	<null>
22	22	2_firstname	<null>	<null>	static	<null>	<null>
23	23	2_lastname	<null>	<null>	static	<null>	<null>
24	24	2_middlename	<null>	<null>	static	<null>	<null>
25	25	2_lastname	<null>	<null>	static	<null>	<null>
26	26	2_prefix	<null>	<null>	static	<null>	<null>
27	27	2_company	<null>	<null>	static	<null>	<null>
28	28	2_street	<null>	Magento\Eav\Model\Entity\Attribute\Backend\DefaultBackend	static	<null>	<null>
29	29	2_city	<null>	<null>	static	<null>	<null>
30	30	2_country_id	<null>	<null>	static	<null>	<null>
31	31	2_region	<null>	Magento\Customer\Model\ResourceModel\Address\Attribute\Backend\Region	static	<null>	<null>
32	32	2_region_id	<null>	<null>	static	<null>	<null>
33	33	2_postcode	<null>	<null>	static	<null>	<null>
34	34	2_telephone	<null>	<null>	static	<null>	<null>
35	35	2_fax	<null>	<null>	static	<null>	<null>
36	36	2_vat_id	<null>	<null>	static	<null>	<null>
37	37	2_vat_is_valid	<null>	<null>	static	<null>	<null>
38	38	2_vat_request_id	<null>	<null>	static	<null>	<null>
39	39	2_vat_request_date	<null>	<null>	static	<null>	<null>
40	40	2_vat_request_success	<null>	<null>	static	<null>	<null>
41	41	1_updated_at	<null>	<null>	static	<null>	<null>
42	42	1_failure_reason	<null>	<null>	static	<null>	<null>
43	43	1_first_failure	<null>	<null>	static	<null>	<null>
44	44	1_lock_expires	<null>	<null>	static	<null>	<null>
45	45	3_name	<null>	<null>	varchar	<null>	<null>

The different values possessed by the attribute will be stored in the data table, the schematic diagram is as follows.

	value_id	attribute_id	store_id	entity_id	value
1	1	45	8	1	Root Catalog
2	2	45	8	2	Clothe
3	3	82	8	2	PRODUCTS
4	4	117	8	2	Furniture
5	5	145	8	2	3
6	6	147	8	2	6
7	7	25	45	8	3 Men
8	8	26	82	8	3 PRODUCTS
9	9	27	117	8	3 men
10	10	28	118	8	3 8
11	11	29	145	8	3 1
12	12	38	147	8	4 Women
13	13	45	82	8	4 PRODUCTS
14	14	98	52	8	4 women
15	15	51	117	8	4 women
16	16	52	118	8	4 women
17	17	53	145	8	4 1
18	18	54	147	8	4 8
19	19	97	137	8	4 Fullwidth
20	20	99	139	8	4 6
21	21	153	45	8	9 All_Clothing
22	22	163	82	8	9 PRODUCTS
23	23	164	117	8	9 adirondack-chair
24	24	165	118	8	9 women/adirondack-chair
25	25	166	145	8	9 8
26	26	167	147	8	9 9
27	27	178	45	8	10 Jackets & Vests
28	28	171	82	8	10 PRODUCTS
29	29	172	117	8	10 teak-wood
30	30	173	118	8	10 women/adirondack-chair/teak-wood
31	31	174	145	8	10 8
32	32	178	147	8	10 8
33	33	179	45	8	11 Hoodies & Pullovers
34	34	179	82	8	11 PRODUCTS
35	35	188	117	8	11 pine-wood
36	36	181	118	8	11 women/adirondack-chair/pine-wood
37	37	182	145	8	11 8
38	38	183	147	8	11 8
39	39	283	45	8	12 Sports Bras
40	40	284	82	8	12 PRODUCTS
41	41	285	117	8	12 cedar-wood
42	42	286	118	8	12 women/adirondack-chair/cedar-wood
43	43	287	145	8	12 8
44	44	288	147	8	12 8
45	45	211	45	8	12 Base Layer

Compared with other database structures, EAV has greater flexibility to allow development of expansion kits, without changing the core database structure, it is easier and faster to add attributes and data.

4.3 API

Magento 2 comes with APIs based on SOAP and REST. There is no longer an API based on XML-RPC. From the perspective of business logic, APIs based on SOAP and REST are equivalent. In addition to providing programmers with a way to programmatically interact with a single independent Magento instance without writing native Magento or PHP code, the Magento 2 REST API is also designed to allow browser-based client-side Javascript code to access API calls. Because this project uses React Native, which is mainly in Javascript, as a mobile application development, Magento 2 REST API is the main API method, so the data obtained through the API will be returned in JSON format.

Magento 2 REST APIs used in this project [3]:

Endpoint	Method	Description
carts/mine	PUT	Save quote
carts/mine	POST	Creates an empty cart and quote for a specified customer if customer does not have a cart yet.
carts/mine	GET	Returns information for the cart for a specified customer.
carts/mine/balance/apply	POST	Apply store credit
carts/mine/billing-address	GET	Returns the billing address for a specified quote.
carts/mine/billing-address	POST	Assigns a specified billing address to a specified cart.
carts/mine/checkGiftCard/{giftCardCode}	GET	Check gift card balance if applied to given cart.
carts/mine/collect-totals	PUT	Set shipping/billing methods and additional data for cart and collect totals.
carts/mine/coupons	GET	Returns information for a coupon in a specified cart.
carts/mine/coupons	DELETE	Deletes a coupon from a specified cart.
carts/mine/coupons/{couponCode}	PUT	Adds a coupon by code to a specified cart.
carts/mine/delivery-option	POST	Handle selected delivery option.
carts/mine/estimate-shipping-methods	POST	Estimate shipping by address and return list of available shipping methods.
carts/mine/giftCards	POST	Add gift card to the cart.
carts/mine/giftCards/{giftCardCode}	DELETE	Remove GiftCard Account entity
carts/mine/items	POST	Add/update the specified cart item.

Endpoint	Method	Description
carts/mine/items/{itemId}	PUT	Add/update the specified cart item.
carts/mine/items/{itemId}	DELETE	Removes the specified item from the specified cart.
carts/mine/order	PUT	Places an order for a specified cart.
carts/mine/payment-information	POST	Set payment information and place order for a specified cart.
carts/mine/payment-information	GET	Get payment information
carts/mine/payment-methods	GET	Lists available payment methods for a specified shopping cart. This call returns an array of objects, but detailed information about each object's attributes might not be included.
carts/mine/totals	GET	Returns quote totals data for a specified cart.
carts/mine/totals-information	POST	Calculate quote totals based on address and shipping method.
products	GET	Get product list
products/cost-information	POST	Return product prices. In case of at least one of skus is not found exception will be thrown.
products/special-price-information	POST	Return product's special price. In case of at least one of skus is not found exception will be thrown.
products/{sku}	GET	Get info about product by product SKU

Endpoint	Method	Description
products/{sku}/media	GET	Retrieve the list of gallery entries associated with given product
products/{sku}/options	GET	Get the list of custom options for a specific product
products-render-info	GET	Collect and retrieve the list of product render info. This info contains raw prices and formatted prices, product name, stock status, store_id, etc.
search	GET	Make Full Text Search and return found Documents
stockStatuses/{productSku}	GET	Get the stock statuses for a specific product

4.4 React Native

4.4.1 LIFECYCLE

React component allows developer to divide the UI into independent and reusable units, and each unit can be drawn to think independently. React component can be defined by inheriting React.Component or React.PureComponent. each component has several life cycle methods, developer can override these methods in order to execute certain programs at specific times during the development process.

Mounting

When an instance of a component is created and added to the DOM, its lifecycle will call these methods in the following order:

- constructor()

```
constructor(props)
```

- A React component constructor will be called before it is mounted.

Usually in React, the constructor will only serve two purposes:

- Initialize the internal state by specifying a `this.state` object.
- Bind an instance to the event handler method.

- `static getDerivedStateFromProps()`

```
static getDerivedStateFromProps(props, state)
```

- `getDerivedStateFromProps` will be called before a component is rendered, whether it is during the first mount or subsequent updates. It should return an object to update the state, or null to indicate that no state needs to be updated.

- `render()`
- `componentDidMount()`

- After a component is mounted (added to the DOM tree), `componentDidMount()` will be called immediately. The initialization that requires DOM node should be written in this method.

Updating

When the prop or state changes, an update will be generated. When a component is re-rendered, its life cycle will call these methods in the following order:

- `static getDerivedStateFromProps()`
- `shouldComponentUpdate()`
- `render()`
- `getSnapshotBeforeUpdate()`

```
getSnapshotBeforeUpdate(prevProps, prevState)
```

- `getSnapshotBeforeUpdate()` will be called when the last render output is submitted to the DOM. It allows to grab some information (such as the position of the scroll axis) from the DOM before it changes. The value returned by this lifecycle method will be passed as a parameter to `componentDidUpdate()`.
- `componentDidUpdate()`

```
componentDidUpdate(prevProps, prevState, snapshot)
```

- `componentDidUpdate()` will be called immediately after the update. This method will not be called during the first render.
- After the component is updated, the DOM can be operated here.
- If the component has `getSnapshotBeforeUpdate()` this rare life cycle method, the value returned will be passed to `componentDidUpdate()` as the third “snapshot” parameter. Otherwise, this parameter will be undefined.

Unmounting

When a component is removed from the DOM, this method will be called:

- `componentWillUnmount()`
 - `componentWillUnmount()` will be called as soon as each component is unmounted and destroyed. Developer can perform any cleanup within this method, such as canceling timers and network requests or removing any subscriptions created in `componentDidMount()`.

Error handling

When a component fails during the render process, life cycle, or in the constructor of a child component, these methods are called:

- static getDerivedStateFromError()

```
static getDerivedStateFromError(error)
```

- This lifecycle method will be called after an error is thrown by a descendant component. It will receive the error as its parameter and return a value to update the state.
- componentDidCatch()

```
componentDidCatch(error, info)
```

- This lifecycle method will be called after an error is thrown by a descendant component. It accepts two parameters:
 - error - the error being thrown.
 - info - An object with a componentStack key that contains information about which component threw an error.
- componentDidCatch () will be called during “commit”, so side effect is allowed.

4.5 Image Classification

TensorFlow.js was released a versions for React Native and Expo applications. It allows developers to load per-trained models and train new models in the mobile application. In this project, TensorFlow.js and MobileNet per-trained model architecture are used for to classification of input images in React Native application.

4.6 UI Design - Mobile Platform

4.6.1 CONNECTED WITH MAGENTO API

This section shows the user interface of the mobile application which call the Magento API and get the data to render.

As given in Figure 4.1(a), our index page will show the banner, discount product, and the hot product.

Product Detail, as given in Figure 4.1(b), will show the product gallery, basic information and the option of the product.

Search page, as given in Figure 4.2(a), uses for search the product by keywords.

Figure 4.2(b) shows the login page.

4.6.2 WITHOUTCONNECT MAGENTO API

This is the user interface of the mobile application with the dump data and for test the development function. The reason why have this version of the applicaiton is because before put one function on the connected API version, we need to see the funtion is there any error or conflict which will affect the API version.

Figure 4.3 shows the index and catalog pages:

Figure 4.4 shows the product detail and search pages:

4.7 UI Design - Web Platform

Figure 4.5, 4.6 and 4.7 show index page, product detail page adn checkout pages of our web platform.

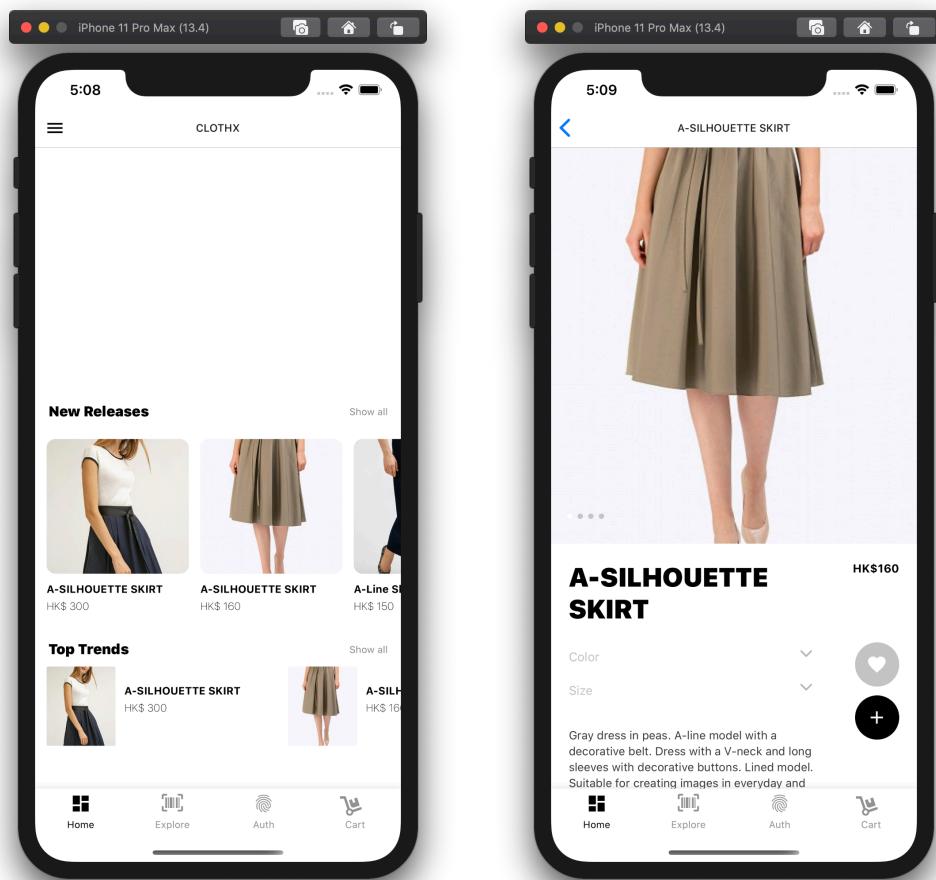


Figure 4.1: (a) Index Page, (b) Product Detail

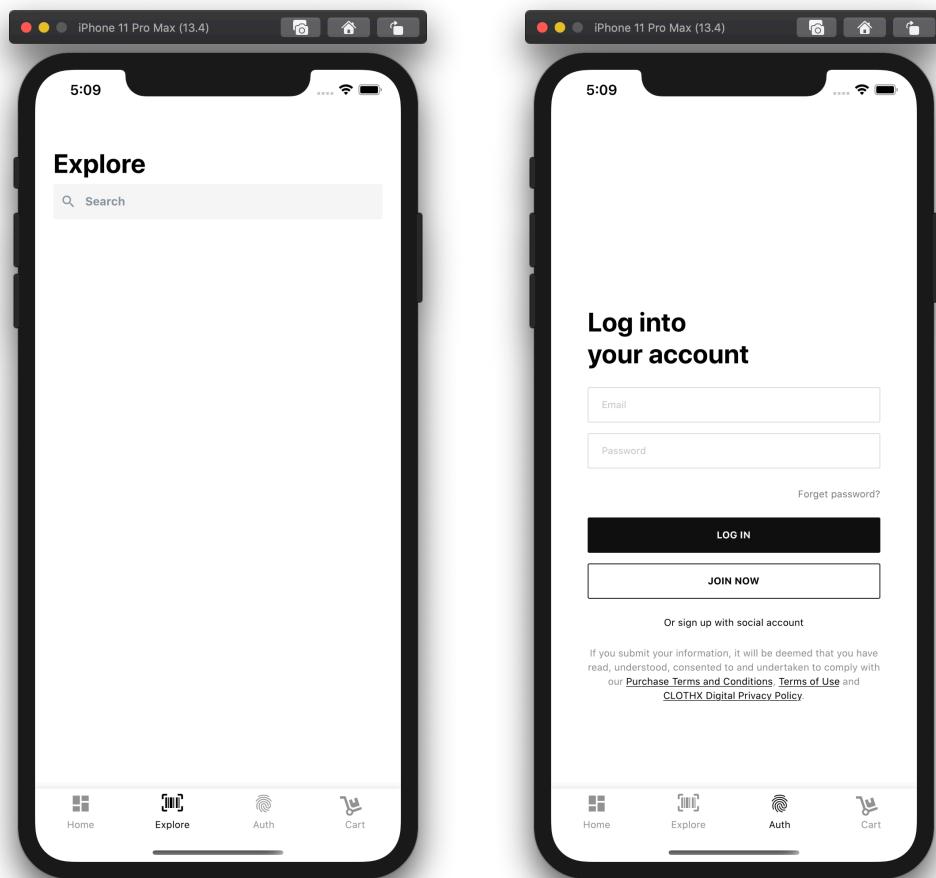


Figure 4.2: (a) Explore, (b) Login

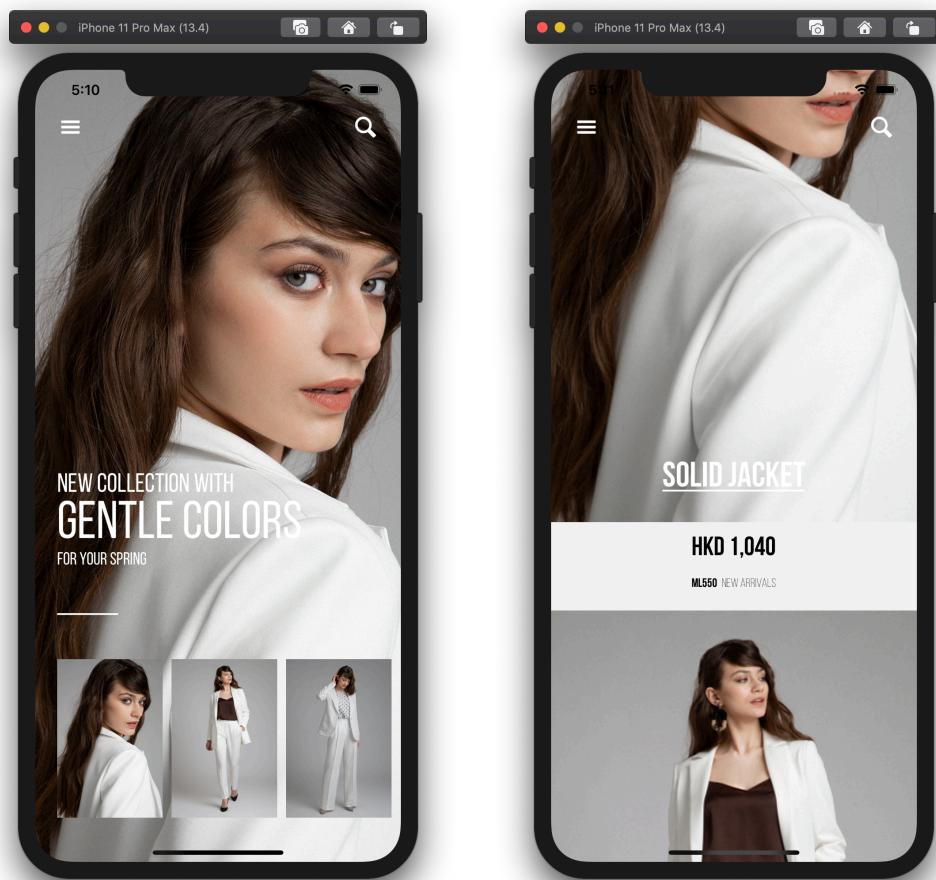


Figure 4.3: (a) Index Page, (b) Catalog

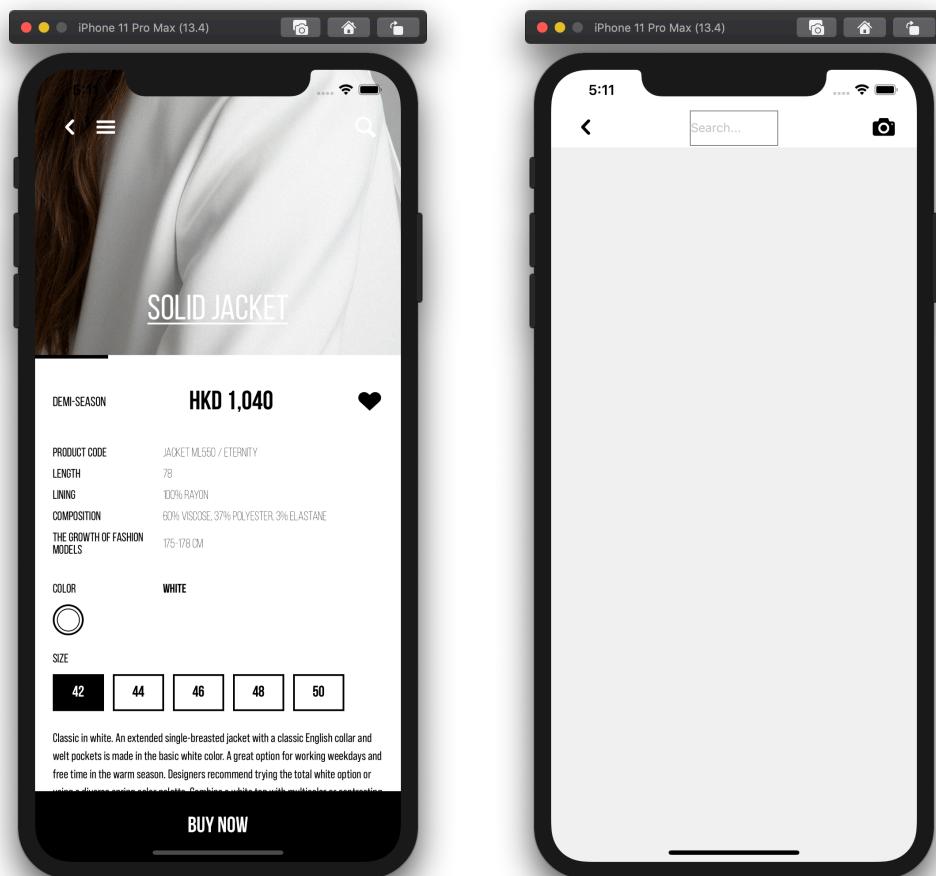


Figure 4.4: (a) Product Detail, (b) Search

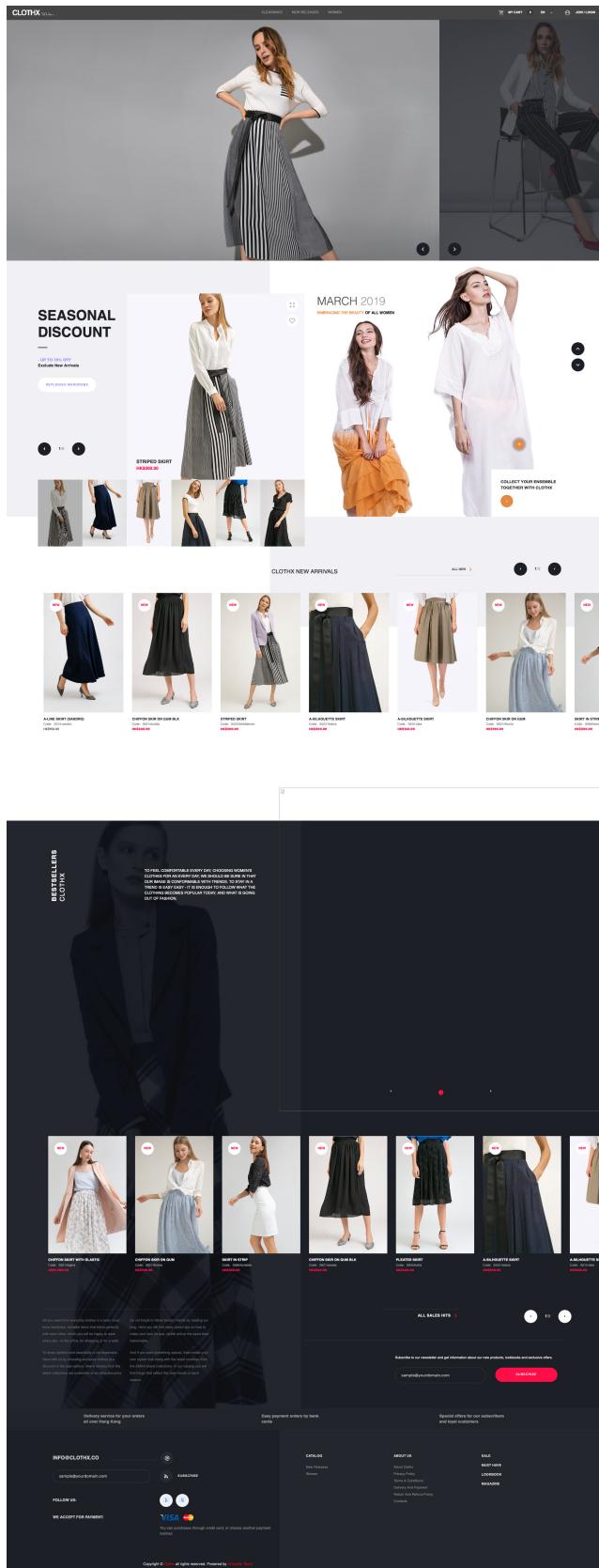


Figure 4.5: Index page

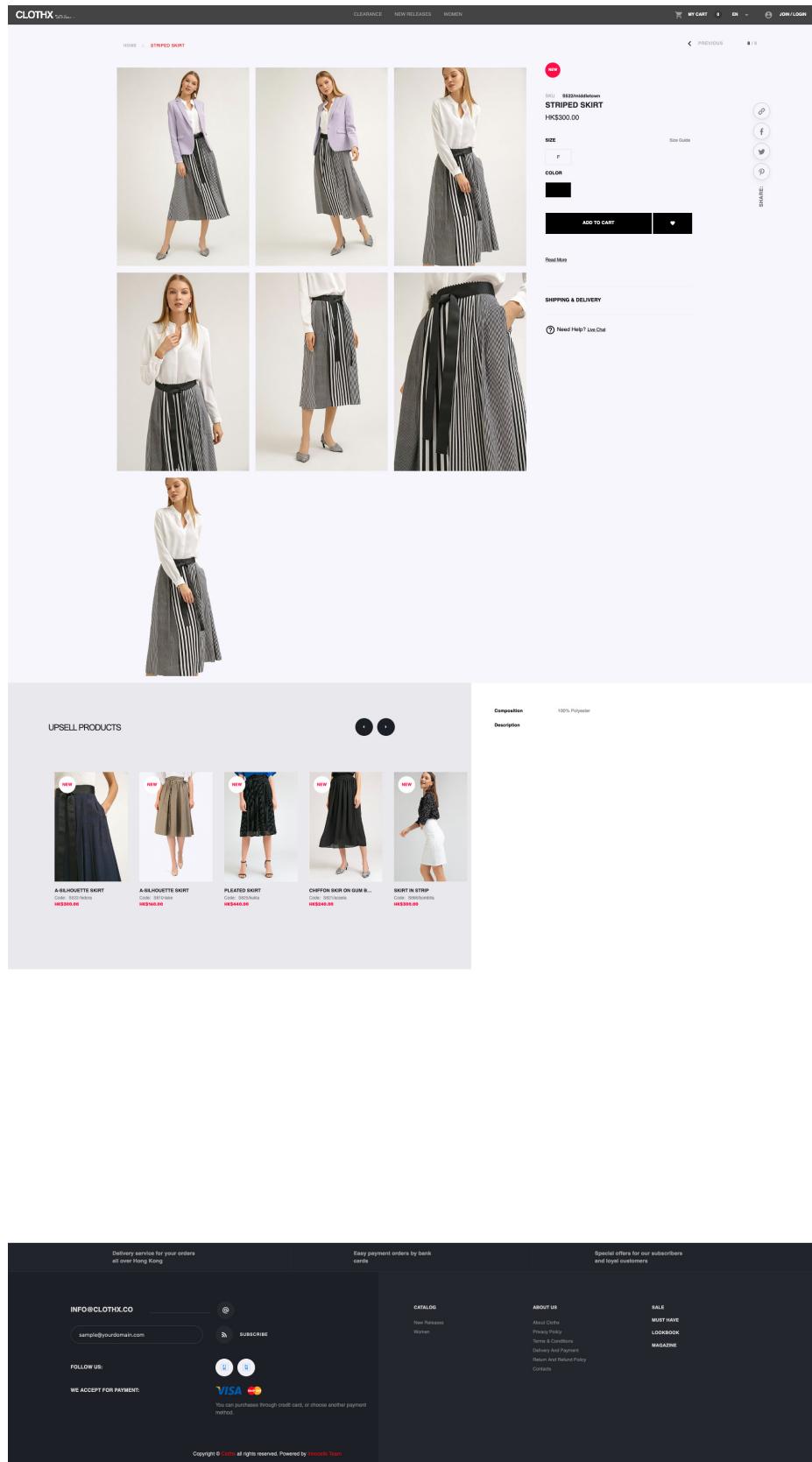


Figure 4.6: Product Detail
31

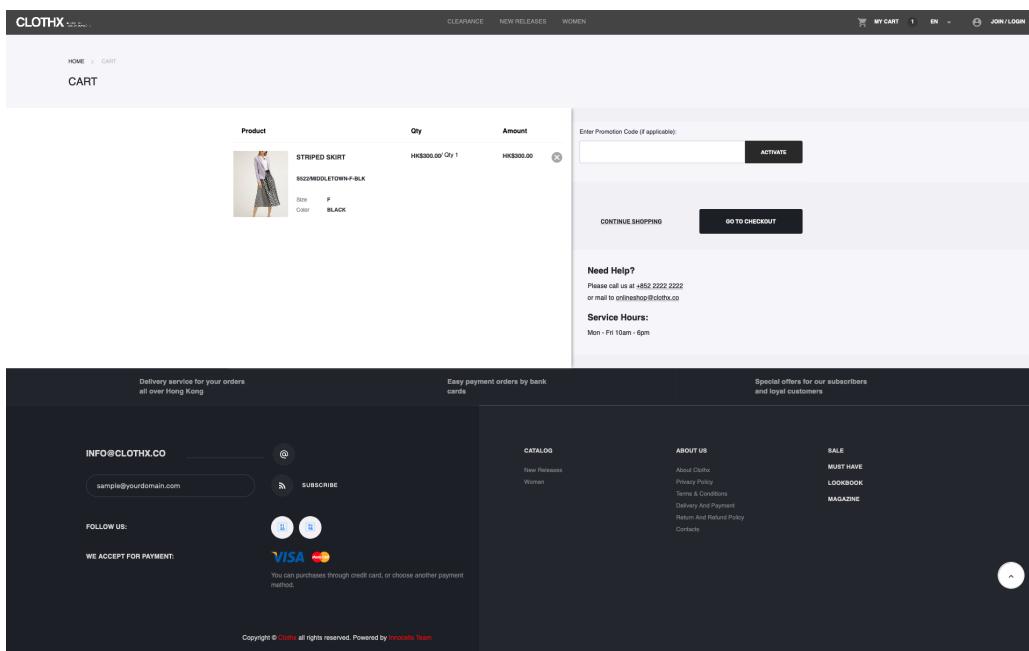


Figure 4.7: Checkout

Chapter 5

Critical Evaluation

Unlike what the project have anticipated from the very beginning, it faced many challenges during the development of the system which led the project to change the strategy in turn of development focus. The project had to leave the web store at its basic function state in order to catch up on the schedule. And over the course of the development, the project have come across some aspect that it have overthought before, like the development of the mobile application of the platform. It is prove that although there are some variation between the initial planning and actually development, it was able to retain most of its intended values and functionalities through flexible implementation.

5.1 Changes to design and justifications

As time moved on while developing the platform, it encountered many different challenges and difficulties. It was able to tackle some of them, like some changes to mobile application design in order to have the mobile application work better with the platform. But some of the challenges were so difficult that we have to straight the project postpone the development of the function, like the in-store purchase, it was unable to use in the latest version of React Native, and due to the upcoming deadline of the project, it had to regrettably put aside its development and focus on the rest of the platform.

5.2 Quality of implementation

As the project did some changes to the design to the overall of the platform, the implementation of the actual platform does a huge different from the previously designed one. For example, the project has to shift the focus entirely away from the building of the web store and cut off the mobile application for designers to ensure the development of the rest of the platform is close to the requirement of the CST student. But overall, most of the users retain their intended functions, the designers are still available to control their settings, and have data analyse service. Customers can browser and search the product from the web store and mobile application. As for the programming techniques, Magento is the main framework for the web application and React Native is used in iOS and Android application.

5.3 Development Difficulties

Since the project time period limitation and the requirement of the CST student, the mainly development in this project will more force on how to use a cross-platform language can be use to built a native like mobile application that can function on both iOS and Android system. Therefore, the web store with Mangeto is only have the basic function that provided by the framework and the development of the mobile application for designers is cut off.

In addition, the version of the React Native I used is the latest version. However, some of the plugins are not yet support the latest one, like use Apple Pay / Google Pay for in-store purchase, therefore it cannot be provided in this demo.

Lastly, the image recognition function need a huge library which have trained. During to the time limited, MobileNet pre-trained model is used for the image recognition. The data which the model provided is not what this project need, therefore it is only for demo usage only.

Chapter 6

Result and conclusion

As a summary to the entire development of the system, it was surely full of challenges and changes. One of the major problems that the project faced was cease of development of the web store and mobile application for designers, it is because as CST student, we were supposed to focus on the development of the software technology portion of the system. The shift of the focus from the system development to technology ensures the build quality of the software technology and makes sure it would be delivered in the schedule that the project planned. There are some limitations to the platform, for example, the platform do not have filter for filter the unrelated information or products which uploaded by designers.

6.1 Future development

1. The control measures function

Since we have to protect the rights of the designer and customers, some function like add the watermark and filter the non related information need to provide in the platform.

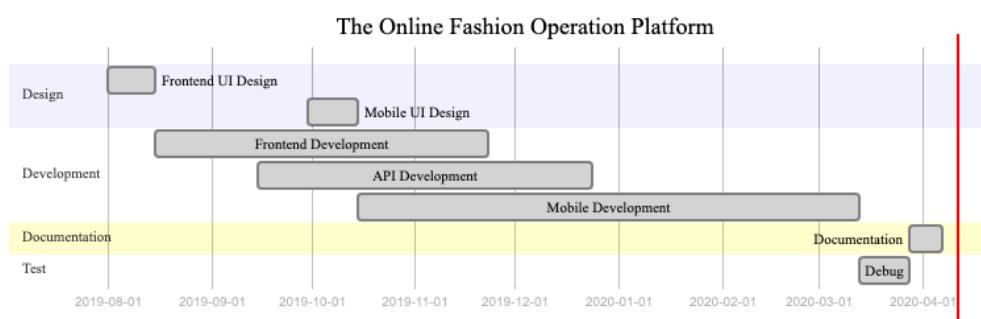
2. Offline support

We can connect with the offline store to let the designers has more way to promote their design and have more data collection.

3. Customised service

Some of the customers are looking for the customised service but not the finished project. It is a chance to let different designers to provide their ideas and even get some resource.

Appendix 1: Project Schedule



References

- [1] I. Blair, “Mobile app download and usage statistics (2019) - buildfire.” BuildFire, Feb. 2019, [Online]. Available: <https://buildfire.com/app-statistics/>.
- [2] T. Simon, “Tech jargon series: PWA vs native vs cross-platform apps.” Medium; Hackages Blog, Apr. 2019, Accessed: Apr. 08, 2020. [Online]. Available: <https://blog.hackages.io/tech-terms-explained-pwa-vs-native-vs-cross-platform-apps-414bccf83c18>.
- [3] “Magento admin rest endpoints - all inclusive (2.3.4).” Magento, [Online]. Available: <https://devdocs.magento.com/redoc/2.3/admin-rest-api.html>.