វិទ្យាស្ថានជាតិប្រៃសណីយ៍ ទូរគមនាគមន៍ បច្ចេកវិទ្យាគមនាគមន៍ និងពត៌មាន

NATIONAL INSTITUTE OF POSTS,TELECOMS & ICT

SCHOOL OF COMPUTER SCIENCE

វែបសាយពាណិជ្ជកម្មអេឡិចត្រូនិចផ្សារណាត់

PSANAT E-COMMERCE WEBSITE

A Thesis

In Partial Fulfillment of the Requirement for the Degree of

Bachelor Program of Computer Science

Submitted By:

Mr. OUCH Kongkea

Under the supervision of:

Mr. CHUOR Porchourng

July 2019

វិទ្យាស្ថានជាតិប្រៃសណីយ៍ ទូរគមនាគមន៍
បច្ចេកវិទ្យាគមនាគមន៍ និងពត៌មាន

មហាវិទ្យាល័យ វិទ្យាសាស្ត្រកុំព្យូទ័រ

គម្រោងសញ្ញាបត្របរិញ្ញាបត្រ

របស់និស្សិត៖ អ៊ូច គន្ធារ

កាលបរិច្ឆេទការពារនិក្ខេបបទ៖ ១៩ កក្កដា ២០១៩

អនុញ្ញាតិអោយការពារគំរោង

ប្រធានវិទ្យាស្ថានជាតិ៖ ____________________

ថ្ងៃទី   ខែកក្កដា ឆ្នាំ ២០១៩

ប្រធានបទ៖	វែបសាយពាណិជ្ជកម្មអេឡិច ត្រូនិចផ្សារណាត់
ព្រឹទ្ធបុរសស្តីទី ៖	បណ្ឌិត សំ សិទ្ធិសេរី
សហគ្រាស៖	ឃ្យូលប៊ីនស្ទូឌីយោ
គ្រូដឹកនាំគម្រោង៖	លោក ជួ ប៉ោឈួង
អ្នកទទួលខុសត្រូវក្នុងសហគ្រាស៖	លោក ស សៀងហុង

រាជធានីភ្នំពេញ

National Institute Of Posts, Telecoms & ICT

School of Computer Science

A Thesis submitted by:

Mr. Ouch Kongkea

Defense Date: 19 July 2019

Thesis defense authorization

President: ________________________

July _____, 2019

Topic:	PSANAT E-COMMERCE WEBSITE
Enterprise:	Coolbeans Studio
Acting Dean:	Dr. Sam Sethserey
Advisor:	Mr. Chuor Porchourng
Supervisor:	Mr. Sor Seanghong

Phnom Penh

ACKNOWLEDGEMENT

First and foremost, I would like to dedicate my wholehearted thank and gratitude to **my mom and dad** for bringing me into this world and raising me up to be who I am today. Without your continuous support and motivation, I would not have made it this far to the point of almost graduating from a prestigious university and having interned at a world-class company.

I would like to express my sincere gratitude to His Excellency **Dr. SENG Sopheap**, President of **National Institute of Posts, Telecoms and ICT**, and other respectful NIPTICT co-founders for essentially founding NIPTICT with a spectacular vision of creating new human resource driven by the application of entrepreneurship and technological innovation for a digitally developed Cambodia.

I would like to specially thank **Mr. CHUOR Porchourng**, Head of the Department of Computer Science and also as my valuable internship advisor, for his tireless effort in directing and effectively coordinating with everyone in the department with the single goal of achieving excellence for both students and lecturers. Whenever I came across obstacle in internship, he was always there to help me find solution by trying to understand the problem from my perspective and then providing valuable advice from his experience. Moreover, I was able to perform work smoothly in an organized manner thanks to the knowledge of software engineering that he passionately imparted to me during the junior year.

I would like to greatly thank **Mr. Sor Seanghong**, senior web developer at Coolbeans Studio, for their invaluable supervising and mentoring me on what it takes to be a professional web developer that succeeds in this competitive technology-driven world. He willingly guided, assigned and trusted me with real projects of the company; thus, I was able to complete the required work in time and be equipped with hands-on knowledge and experience that are useful for my future career.

Last but not least, tremendous thanks and greetings to my fellow **CS4** classmates and lecturers for your support and gift of knowledge that have shaped me into who I am today.

មូលន័យសង្ខេប

និក្ខេបបទនេះរៀបរាប់អំពីការងាររបស់ខ្ញុំក្នុងកំឡុងពេលកម្មសិក្សារយៈពេលបួនខែនៅក្រុមហ៊ុនឈ្មោះ Coolbeans Studio ក្នុងនាមជាអ្នកអភិវឌ្ឍវែបសាយ។ កម្មសិក្សានេះគឺជាផ្នែកមួយដ៏សំខាន់នៃកម្មវិធីសិក្សានៃការបញ្ចប់ថ្នាក់បរិញ្ញាបត្រវិទ្យាសាស្ត្រកុំព្យូទ័រនៅ NIPTICT ។ កម្មសិក្សានេះបានចាប់ផ្តើមពីថ្ងៃទី ១ ខែមីនារហូតដល់ថ្ងៃទី ៣០ ខែមិថុនាឆ្នាំ ២០១៩ ។ ប្រធានបទនៃកម្មសិក្សាគឺ វែបសាយពាណិជ្ជកម្មអេឡិចត្រូនិចផ្សារណាត់ (PSANAT E-COMMERCE WEBSITE) ដែលជាវែបសាយប្រើប្រាស់ទំនើបនិងត្រូវបានបង្កើតឡើងសម្រាប់អនាគតក្រុមហ៊ុនពាណិជ្ជកម្មអេឡិចត្រូនិចមួយ ដែលផ្តោតលើការលក់ផលិតផលខ្មែរក្នុងស្រុក ដោយផ្តោតសំខាន់លើការផ្តល់បទពិសោធន៍ប្រើប្រាស់ងាយស្រួលជូនអតិថិជន។ ដូច្នេះ ខ្ញុំបានចូលធ្វើការនៅក្រុមហ៊ុន Coolbeans Studio ជាអ្នកអភិវឌ្ឍវែបសាយពេញម៉ោងនៅដើមខែមីនា ហើយខ្ញុំត្រូវបានចាត់តាំងគម្រោងនេះ ព្រមទាំងទទួលទទួលគម្រោង និងសេចក្តីលំអិតពីរៀមច្បងរបស់ខ្ញុំដែលគាត់មានឈ្មោះថាលោកសសៀងហុង។ ទំនួលខុសត្រូវចំបងរបស់ខ្ញុំគឺការអភិវឌ្ឍប្រព័ន្ធទាំងមូល ដែលក្នុងនោះរួមមានការពាក់ព័ន្ធនឹងការបង្កើត API ចាំបាច់តាមរយៈការប្រើ Parse Server ដែលដំណើរការលើ Node.JS ហើយបន្ទាប់មកទៀតយក API ទៅអនុវត្តលើការអភិវឌ្ឍនៃផ្នែកខាងមុខនៃវែបសាយអេឡិចត្រូនិចដោយប្រើប្រាស់ Nuxt.JS និងផ្នែកគ្រប់គ្រងដោយប្រើ Angular។ ក្នុងអំឡុងពេលនៃគម្រោងនេះ ខ្ញុំបានប្រើប្រាស់ Node.JS ដល់កម្រិតកំពូល តាមរយៈការប្រើប្រាស់មុខងារទំនើបនៃ Parse Server រួមមានដូចជាមុខងារ Cloud Code មុខងារ Cloud Functions និងមុខងារ Cloud Trigger ហើយថែមទាំងបានពង្រីកយ៉ាងរលូនលើជំនាញអភិវឌ្ឍផ្ទាំងមុខនៃវែបសាយដែលខ្ញុំខ្វះខាតពីមុនមកដូចជា Angular និងចាប់យកបច្ចេកវិទ្យាថ្មី Nuxt.JS ដែលមានសមត្ថភាពខ្លាំងសម្រាប់ការអភិវដ្ឍកម្មវិធីមានភាពទូលំទូលាយ និងរួចរាល់សម្រាប់ប្រើក្នុងផលិតកម្ម។ លើសពីនេះទៅទៀត ខ្ញុំបានទទួលបានបទពិសោធន៍ផ្ទាល់លើការអនុវត្តវិធីសាស្ត្រ Scrum ក្នុងការងារអភិវឌ្ឍកម្មវិធី ហើយទទួលបាននូវជំនាញទន់ដ៏មានអត្ថប្រយោជន៍ជាច្រើន ដែលមានសារៈសំខាន់សម្រាប់ការរីកចម្រើនវិជ្ជាជីវៈរបស់ខ្ញុំ។ ជាការពិតណាស់ ខ្ញុំបានប្រឈមនឹងឧបសគ្គមួយចំនួនក្នុងអំឡុងកម្មសិក្សា ដូចជាសម្ពាធនិងកង្វះចំណេះដឹងផ្នែកបច្ចេកទេស ប៉ុន្តែទោះបីយ៉ាងណាក៏ខ្ញុំបានតស៊ូជំនះលើឧបសគ្គទាំងនោះដោយទទួលជោគជ័យ តាមរយៈការខិតខំប្រឹងប្រែងស្វែងរកដំណោះស្រាយប្រកបដោយភាពច្នៃប្រឌិត និងរៀនអ្វីថ្មីៗ រួមជាមួយការសម្របខ្លួនខ្ញុំទៅក្នុងបរិយាកាសការងារនិងការទាក់ទងជាមួយអ្នករួមការងារ។ នៅចុងបញ្ចប់នៃកម្មសិក្សា ខ្ញុំបានបញ្ចប់រាល់ការទទួលខុសត្រូវសំខាន់ៗដែលបានចាត់តាំងឲ្យខ្ញុំ ហើយខ្ញុំសង្ឃឹមថា វែបសាយពាណិជ្ជកម្មអេឡិចត្រូនិចនេះនឹងផ្តល់នូវលក្ខណៈងាយស្រួលជារៀងរាល់ថ្ងៃដល់អ្នកទិញឥវ៉ាន់នៅស្រុកខ្មែរ និងចូលរួមផ្លាស់ប្តូរវិជ្ជមានដល់ប្រព័ន្ធអេកូឡូស៊ីនៃពាណិជ្ជកម្មអេឡិចត្រូនិចនៅកម្ពុជា។

ABSTRACT

This thesis describes my work during the four-month internship at Coolbeans Studio as a junior web developer. This internship is a crucial part of the Bachelor of Computer Science's curriculum at NIPTICT. The internship started form 1st-March to 30th-June 2019. The topic of the internship is PSANAT E-COMMERCE WEBSITE, a state-of-the-art website built for an upcoming e-commerce company focused on providing the sales of local Khmer products with seamless customer experience. So, I joined Coolbeans Studio as a full-time developer in early March and I was immediately assigned with this project by my senior web developer, Mr. Sor Seanghong, and provided with requirements for this project. My main responsibility was full-stack development of the system which involves creating the necessary API using Parse Server that runs on Node.JS and apply the API onto the developing of the e-commerce frontend using Nuxt.JS as well as the admin side using Angular. Over the course of this project, I managed to harness the full power of Node.JS ecosystem by diving deeper into advanced features of Parse Server such as Cloud Code, Cloud Function and Cloud Trigger and also improve exponentially on the frontend skill I previously lacked in such as Angular and grab new frontend technology called Nuxt.JS which is great for building production-ready and universal Vue application. On top of that, I experienced realistic Scrum Methodology being applied to development work and gained many useful soft skills which are vital to my professional growth. Indeed, I faced some challenges during the internship such as pressure and lack of technical knowledge but I successfully overcame them by consistently working hard to find creative solution and learn new things and adapting myself to the work environment and diverse communication. By the end of the internship program, I have completed all my assigned responsibility and I hope that this ecommerce website will provide convenience to Cambodian shoppers everyday and disrupt the e-commerce ecosystem of Cambodia.

ABBREVIATIONS AND SYMBOLS

HTML	:	Hyper Text Markup Language
CSS	:	Cascading Style Sheets
JS	:	JavaScript
API	:	Application Program Interface
JSON	:	JavaScript Object Notation
NIPTICT	:	National Institute of Post, Telecoms & ICT
GUI	:	Graphical User Interface
UX/UI	:	User Experience/User Interface
Node	:	Node.js
Express	:	Express.js
Nuxt	:	Nuxt.js
Angular	:	Angular 8
Parse	:	Parse Server
SDK	:	Software Development Kit
CLI	:	Command Line Interface
MVC	:	Model-View-Controller
SEO	:	Search Engine Optimization
CRUD	:	Create, Read, Update, Delete

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

1. Presentation of the Internship

1.1. Objective

Having studied computer science at NIPTICT for four years, students are supposed to have gained a lot of practical skills and experience which are applicable for real-world enterprise work. Thus, NIPTICT requires the year-four students to utilize what they have learnt and conduct a four-month internship at a standardized and professional company. During this period, the goal is for them to realistically apply their talent and unleash their potential to contribute towards actual company project as well as grasp professional manners such as communication and work ethics and then get ready for the competitive world. This internship also essentially serves as a bridge to the success of future career because after the completion of the internship they are required to hand in their thesis and defense regarding their internship in order to officially pass through the final year and graduate with bachelor degree of computer science.

1.2. Duration

The internship at Coolbeans Studio started from 1st March 2019 and ended on 30th June 2019, so the duration is approximately 4 months which suits the internship's requirement.

2. Presentation of the Organization

2.1. Background

Coolbeans Studio [5] is a rising-star custom software design and development firm based in Phnom Penh with the focus on assisting organizations of all size to bring their products to reality. We are a proud team in the core aspect of building products with the intention of successfully becoming a product-market-fit and a viable business. Our main focus within the Cambodia tech industry is developing scalable, robust, and technically sophisticated custom web, mobile, and e-commerce applications, based on platforms operating on top of cloud infrastructure.

Figure 1: Coolbeans Studio Logo

Coolbeans aspires not to only build a solid technical solution but also work with our client in developing a product that is closely aligned with their essential business need and target user experience, which results into significant reduction of cost and growth in business profitability and customer number.

2.2 Services

Multi-platform mobile application for your business? Responsive website to show off your company? Or beautiful graphics to promote your team? We got you covered. We are providing a range of services from Mobile to Web application, while keeping our eyes on those beautiful and modern design principles.

Figure 2: Mobile App Service

Mobile application is always our core service because we know it's important for your business. Native or Hybrid? We got you covered.

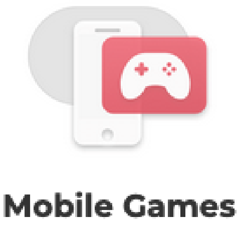


Figure 3: Mobile Games Service

We love building games just as much as we love playing them (oops). We have been working with several 2D games, and 3D one is our next target.

Figure 4: Web Development Service

If you have web browser, you can browse the web. That's how convenient it is, and that's why we believe your company will need it.

Figure 5: UX/UI Design Service

We love beautiful design and we know you do too; that's why we care so much about our design principle. We want to make your product stand out from the rest.

Figure 6: E-commerce Service

As more and more people get connected to Internet, the chance to sell your products online is becoming bigger and bigger. Having an e-commerce site done right, your customer will be able to engage better with your business.

3. Address and Contact

Having a big idea that you want to start? Let's talk. Coolbeans Studio is reachable through the following means:

- ❖ Address : 206 Eo Preah Norodom Blvd (41), Sangkat, Tonle Bassac, Khan Chamkamorn, Phnom Penh
- ❖ Email : info@coolbeans.studio
- ❖ Telephone : 888 / (+855) 17 789 742
- ❖ Website : www.coolbeans.studio

4. Organization Chart

Coolbeans was found by talented veterans from DMI who came together to create their own firm with the vision of Cambodian digital solutions being localized based on human-centered design inspired by both Khmer and exotic style. Our team works together closely and each of us has multiple roles since we are a rising startup. We also have partnership and connections with other firms as well in case we need more workforce.

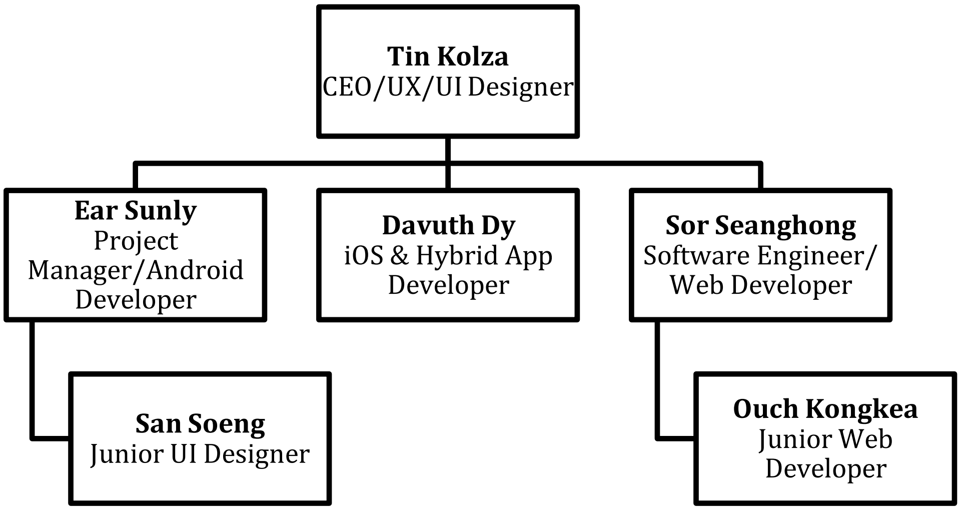


Figure 7: Organization Chart

II. PRESENTATION OF THE PROJECT

1. General Presentation of the Project

The title of the project is “PSANAT E-COMMERCE WEBSITE” which is one of the projects I involved in during the 4-month internship. It is a state-of-the-art website built for an upcoming e-commerce company focused on providing the sales of local Khmer products with seamless customer experience.

2. Problematic

There is a number of problems that leads to the development initiative of this project. First, in term of e-commerce ecosystem in Cambodia, most e-commerce websites lack trust, diversity of local and foreign products and cashless payment adoption. Second, in term of technology, most e-commerce websites only use server-side rendering due to development convenience but it comes at the cost of user experience. Last but not least, in term of user experience, nowadays e-commerce websites still lack in human-centered design and intuitiveness that are what attract and shape good user journey.

3. Objective

PSANAT aims to be the leading B2C e-commerce website in Cambodia that establishes a clear information architecture, include compelling content, offer details about products and services that users care about, and have simple, understandable interaction design based on the advantages of smooth client-side and server-side browsing as well as optimization for SEO. The main purpose is to bridge the gap between great user experience and seamless browsing performance by prioritizing provision of relevant and quality local and foreign products as well as partnering with local and foreign supply chains to deliver the best affordable products for consumers.

4. Development Methodology

To effectively initiate a project, it is essential to carefully choose a well-suited and effective type of methodology. Hence, Scrum methodology was specifically selected due to the nature of its flexibility and ease in adoption method as it supports a great variety of different types of project.

Agile Software Development with Scrum is usually considered as a methodology. In such methodology, the project manager plays the role of the Scrum Master. Each scrum lasts the period of one week and we carry out a small meeting about the process of each developer for approximately 10 minutes every morning.

Each task was assigned by project manager and was split into sprint in total that one sprint took about 2 weeks. We always have our stand-up meeting every morning around 5-10 minutes to discuss about our previous tasks, challenges, solutions, struggles and the planning for upcoming tasks.

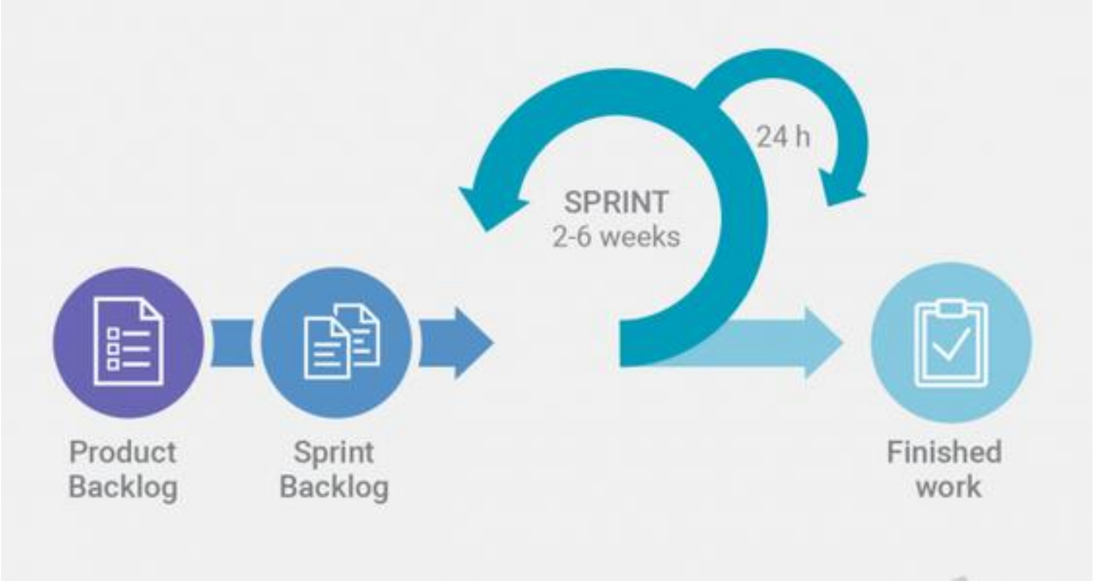
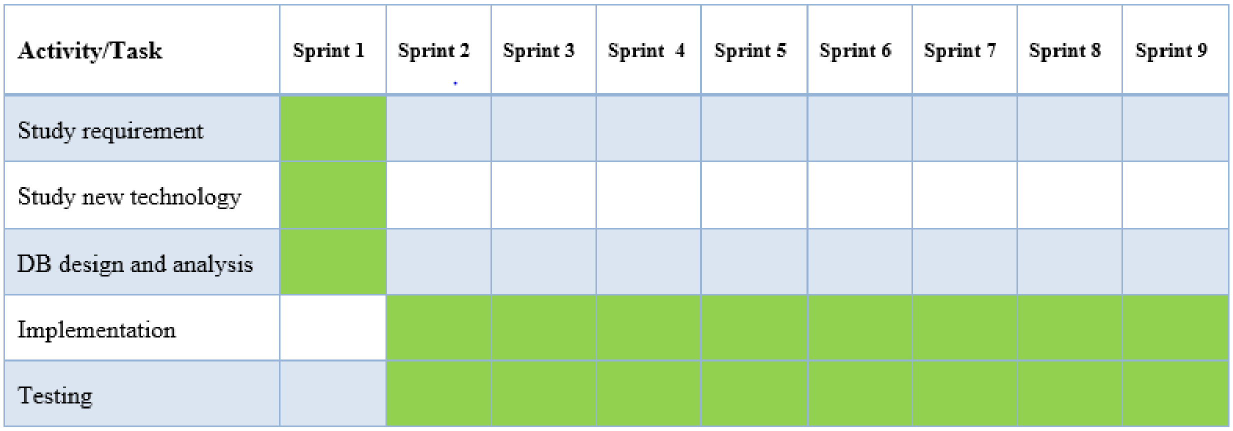


Figure 8: Scrum Methodology

5. Project Schedule

The following table demonstrates my plan during the 17 weeks of PSANAT e-commerce project that uses Scrum as I have described in section 4.

First week and second week, I spent on study requirement and learning technology and it was before internship starts but I already joined the company back then. Then the implementation started spanning from week 3 to week 17 that was spitted into 8 sprints in which 1 sprint last 2 weeks as I mentioned in development methodology. At the end of each sprint, there’s testing with product owner to find flaws in the features done in the sprint so that the project manage can reassess what to fix and what to include in the next sprint’s backlog. This saved so much money and development time as we found out bugs and got new feature requests from product owner earlier and by the final sprint, the system is ready and mature to be deployed on final production.



Activity/Task	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9
Study requirement	🟩								
Study new technology	🟩								
DB design and analysis	🟩								
Implementation		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Testing		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Table 1: Project Schedule

6. Task breakdown

The table below shows the tasks that we have divided, which is specifically the work breakdown structure in our project development. There are two people in the development team which is me and my senior developer, Mr. Sor Seanghong, so the project manager split the tasks accordingly based on our expertise.

Client Side + API	**Role**
User Signup with Firebase and Email	Sor Seanghong
User Login with Firebase and Email	
Password Reset with Firebase and Email	
Account Verification with Firebase and Email	
Subscription with Email	Ouch Kongkea
Order Confirmation Email	
Log User’s Activities	
Log Product View by User	
Product Search	
Hot Search Keyword	
User Personalized Products	
Product Listing	
Product Details	
Checkout with Cash On Delivery	

Manage Profile	
Order Tracking	
Order Overview	
Order Dashboard	
Manage Address	
Review Ordered Product	
Review Dashboard	
Render Meta Tags Dynamically	
Banner and Promotions	Seanghong Sor
Checkout with ABA PAY	
Top Categories	
Spotlight	
Cart	
About Us	
Contact Us	
Online Payment Policy	
Admin Side + API	**Role**
Sale Report Dashboard	Ouch Kongkea
Top Sale Products	
Top Viewed Products	
Log User’s Activites	
Manage User	
Manage Order	
Manage Product	Sor Seanghong
Manage Banner	
Manage Shipping Zone	
Manage Category	
Manage Stock	
Manage Static Page	
Manage Hot Keyword	Ouch Kongkea
Manage Subscription	

Manage Meta Tags for Product and Static Page	
Manage Configuration	
Export to Excel (for all manage features + user log)	

Table 2: Task breakdown

III. ANALYSIS AND GENERAL CONCEPT

1. Requirements

To get a better picture of the objective of the project, we need to list out and briefly elaborate each of all the requirements of the project.

1.1. Functional Requirements

The functional requirements are the core functionalities needed to be implemented in order to make our web application works. The details of these functionalities are as following:

User Feature	Requirement
Authentication	• User can sign up, login, verify account and reset password with phone number and email address. • Account verification and reset password should send SMS/email directly to perform such function.
Front page	• User can view various collections of products such as banner, spotlight and category • User gets suggested related products based on product view history • User can see trending product search keywords • User can view static pages that are dynamically managed from database, such as about us, contact us and site policy. • User can subscribe to newsletter by email address. • Product::
Product	• User can search and filter products by price range, brand, category as well as sort by rating, popularity and sale. • User can view product details that include title, rating, images, available stock and description. • User can see their own recently view products and related products in the same category of the currently view product. • User can buy now immediately or add to cart for later checkout
Cart	• User can view and edit cart by just hovering over its icon • Cart should contain each item's image, name and price as well as the total item cost. • User can click checkout from cart

Checkout	• User can still edit items in cart here • User is required shipping information including phone number, email, city/province and specific address. • User can choose cash on delivery and pay later or direct cashless payment through PayPal or ABA Pay. • User should receive order confirmation email after order completion
Profile	• User can create multiple shipping addresses and set default one for checkout as well as delete or update later. • User can enter or update full name, phone number, email address, gender, profile photo and date of birth. • Review: • User can check past product reviews they have made in one place • User should also see their own review on product that they have purchased.
Review	• User can check past product reviews they have made in one place • User should also see their own review on product that they have purchased.

Table 3: Client Requirement

Admin Feature	**Requirement**
Dashboard	• Admin can filter by date range and see summary report of sale performance accordingly based on the filter • Admin can see top sold and top view products.
Administration	• Admin can monitor user activities including signup, login and logout and perform essential search and filter by date range and type of activity to find target. • Admin can perform CRUD on user’s information and perform essential search and filter by date range to find target.

	- Admin can disable to prevent a user from logging in the system. - Admin can perform CRUD on product, hot keyword, page, banner, category and configuration - Admin can view all orders and decide to accept, reject or complete the order as well as and perform essential search and filter by date range, order status, username to find target. - Admin can edit information of stock and shipping zone. - Admin can view subscription email and perform a search on it. - Admin can provide meta tag for product and information page.

Table 4: Admin Requirement

1.2. Non-functional Requirements

Apart from the functional requirements, it is crucial to consider as well about the non-functional requirements as it would contribute largely to the enhancement of the application quality and usability of our system. The non-functional requirements include:

Non-functional Feature	**Requirement**
SEO Support	The client application must use universal framework that can strongly support SEO without compromising performance.
UX/UI	The UX/UI must be simple, intuitive and user-friendly so that users can navigate on our system smoothly and painlessly.
Performance	All steps in developing the new intranet must always give regard to the impact of performance because it is vital to make our system run quickly and efficiently without any errors.

Maintainability	The system must be designed and programmed in a way that is easy and hassle-free for the next developers to maintain.
Scalability	The system architecture must be carefully designed in regard to the scalable nature so that new changes or adjustments can be performed in the future seamlessly and effectively.

Table 5: Non-functional Requirements

2. Analysis of the Project

2.1. System Users

In this application, there are only 3 types of users which are:

Type of User	**Description**
Visitor	can do all the regular stuffs without having to authenticate such as browsing and viewing public pages and products.
User	can perform important ecommerce activities mainly ordering and checking out for the products they want to purchase as well as updating personal information.
Administrator	has both privileges of visitor and user and moreover can perform CRUD operations to manage ecommerce site as well as view sale report.

Table 6: System Users

2.2. Use Case Diagram

There is a wide range of activities available to perform on the system but not all of them are available to any user. There are some inheritance and dependency among users as well as activities which is not surprising due to the diverse and complex nature of our system.

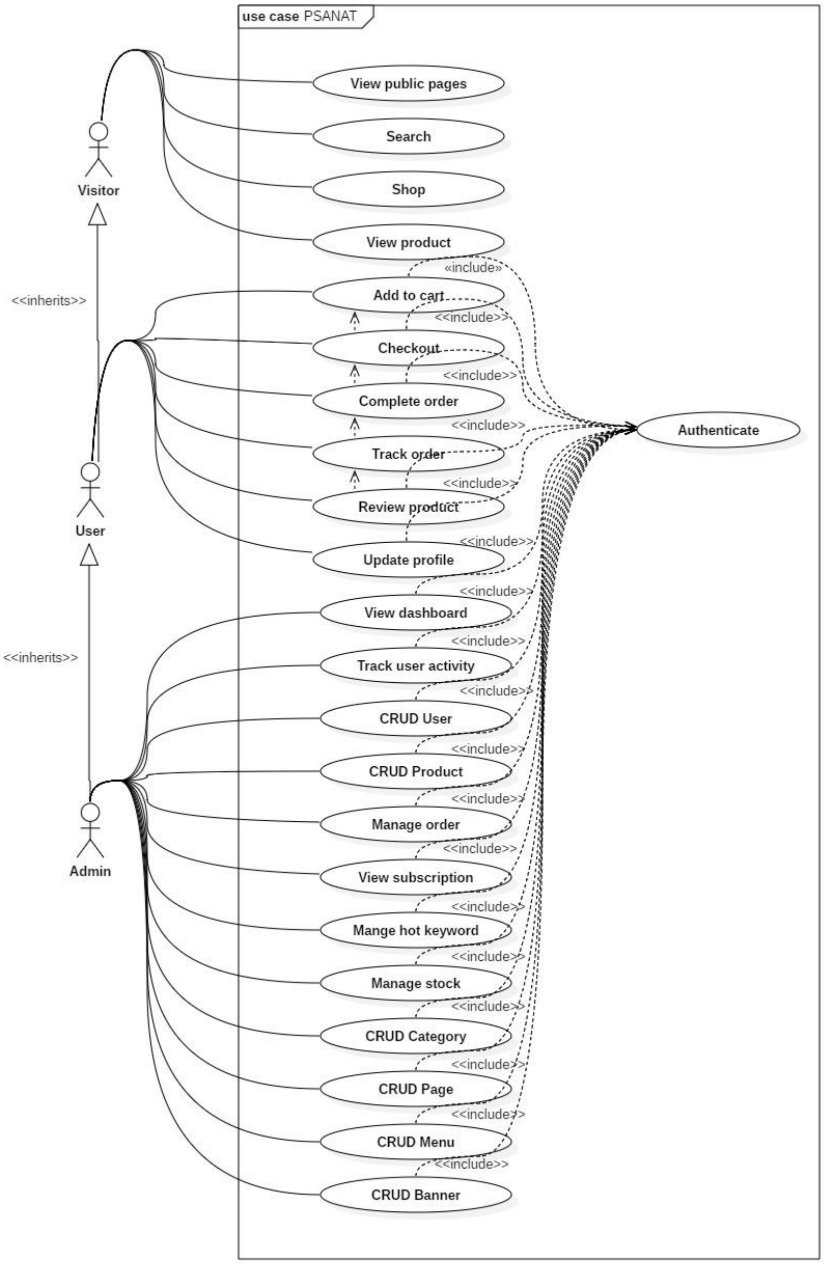


Figure 9: Use Case Diagram

Visitor is the most basic type of user as visitor can access public content such as static pages and product collections and perform activities such as search, shop and view products.

Customer, or simply referred as user, can perform more core ecommerce activities on top of visitor's activities. After having browsed and found their desired products, user can add those products to cart which starts a chain of dependent events. To enter the checkout process, user needs to complete adding products to cart and to complete and place the order, the user needs to have done the checkout steps such as providing shipping info and payment. Then, user can start tracking their order until their order arrives and after that be able to review the products they order. Last but not least, user can update their profile information independently without being required to complete other checkout or order related activities. But, keep in mind, to perform all these activities I have mentioned, the user is first required to authenticate as user first which is the fundamental step.

Administrator, or simply called admin, is the most powerful user of all users. Admin is the one who performs CRUD on the website's content as listed in the diagram and can track any user activity. Crucially, the admin has to approve or reject the customer's order and analyzes the business information through sale dashboard to make important business decision. But, again like customer, admin must authenticate with admin permission first before being allowed to perform any admin or normal activities.

2.3. Activity Case Diagram

This activity diagram elaborates the essential ecommerce experience which is performed by customer and involved PayPal API and PSANAT admin. It explores the customer journey of a successful product purchase on our website and how it interacts with other components.

Just as I have explained in use case diagram, user first needs to browse and add favorite products to cart and start the checkout process. In checkout, user is required to complete two main activities which are providing their shipping information and payment method. If the user select cash on delivery as payment, they can proceed to complete and place the order which is later notified to admin. But, if PayPal option is selected, PayPal transaction will be set up and a window popup will require user to login to their paypal account and pay for the order total cost. When the transaction is completed, the website will receive response from PayPal API to confirm the validity of transaction and user will be allowed to place the order with payment status set to complete.

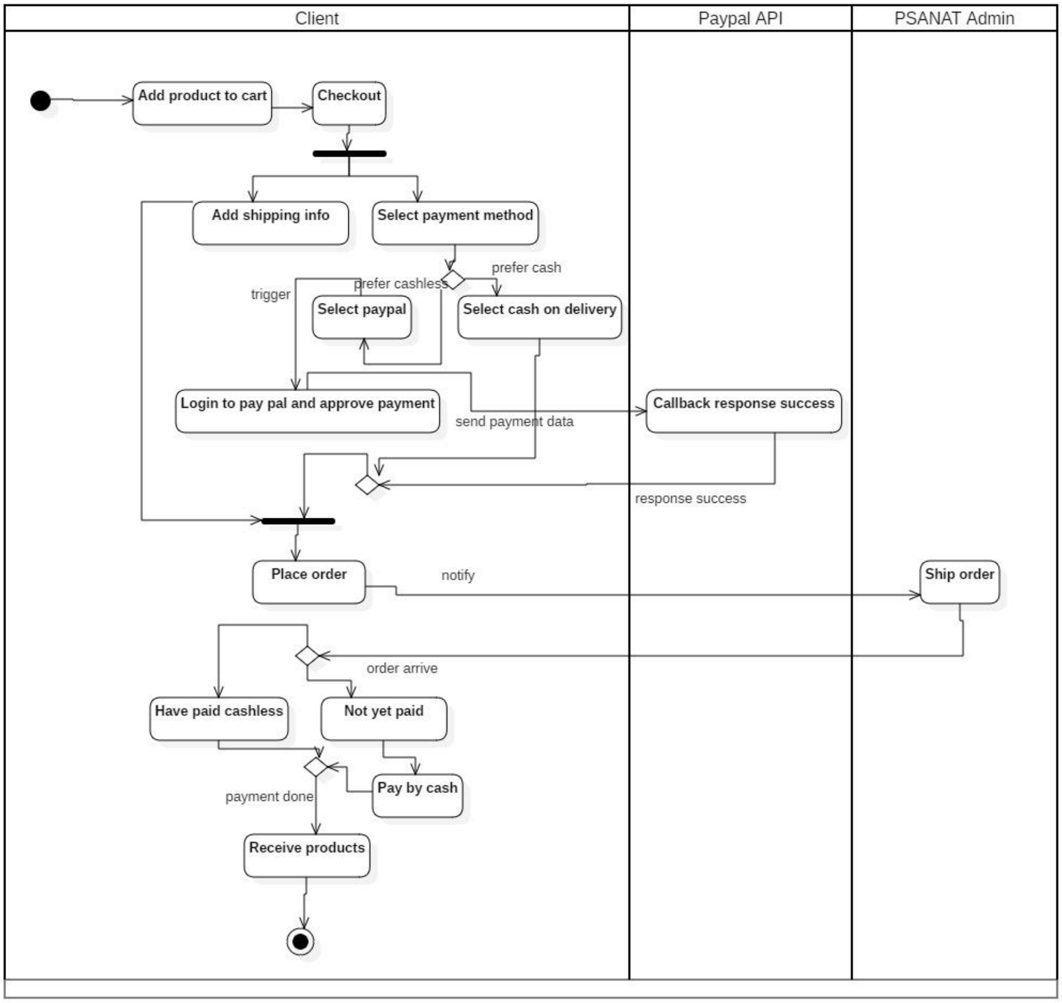


Figure 10: Activity Diagram

When the admin has received and deemed the order as acceptable, the order will be shipped to the user. Upon the order arrival, user doesn't need to pay again if cashless payment has already been given but for user who hasn't paid yet and chosen cash on delivery before, they need to pay now. After the payment has been verified as complete, the user will be able to receive the products they order and the journey of ordering product comes to an end.

2.4. Database Design

Since we use MongoDB, unlike SQL databases, where we must determine and declare a table's schema before inserting data, MongoDB's collections, by default, does not require its documents to have the same schema.

The documents in a single collection do not need to have the same set of fields and the data type for a field can differ across documents within a collection.

To change the structure of the documents in a collection, such as add new fields, remove existing fields, or change the field values to a new type, update the documents to the new structure.

This flexibility facilitates the mapping of documents to an entity or an object. Each document can match the data fields of the represented entity, even if the document has substantial variation from other documents in the collection.

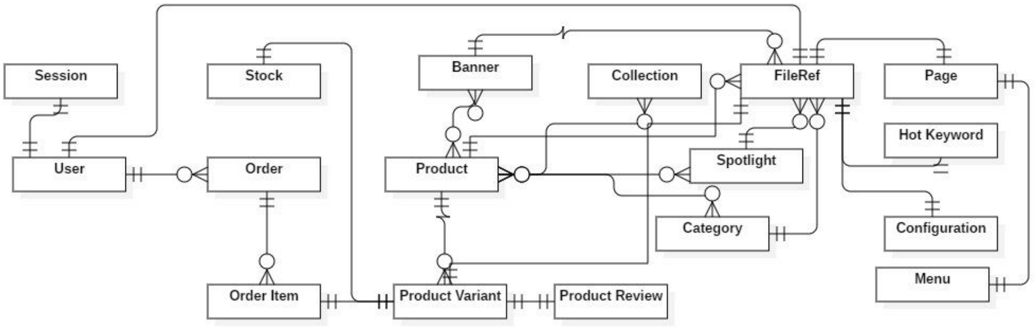


Figure 11 : ER Diagram

Hence, despite the fact that MongoDB is non-schema and capable of simple document storing without complex relational relationship, ER diagram is the most suitable to explain the simplistic structure of our noSQL database.

Simply to how activity and use case diagram’s order process, the database is built around order and product.

One user can make many orders but one order can only belong to one user who creates it. Order contain many order items linking to specific product variants which refer to main product. One product can have many variants and can be included many in product collections such as banner, spotlight and category to display on homepage. This collection also can contain many photo files which link to FileRef (File Reference). Page, hot keyword and configuration contain many file references the same but footer menu can only be linked to only one page. Stock is related one-to-one with product variant in order to provide current stock number of product variant. The same goes for product review, which is connected one-to-one with product variant as well.

IV. DETAIL CONCEPT

1. Choice of Technologies

In choosing technologies for our ecommerce project, we mainly consider system scalability, developer expertise and the requirements of our product owner. The cost of developing and maintaining the software should be reasonably priced and the development of software should be smoothly completed in the specified time-frame.

Figure 12: Nuxt.js Logo

Nuxt.js [1] framework is the perfect candidate that helps us build server-rendered Vue.js applications easily. It abstracts most of the complex configuration involved in managing things like asynchronous data, middleware, and routing. It’s similar to Angular Universal for Angular, and Next.js for React. Nuxt also supports Typescript which is mainly used by our developers across Node.js frameworks and libraries for clean code organization and maintainability.

Nuxt.js meets our considerations and stands out to be the most suitable choice for ecommerce client site due to its universality and its rich set of advantages.As client focuses on development speed and SEO support without sacrificing quality and scalability of the application.

Figure 13: Angular Logo

Angular [2] comes with almost everything we need to build a complicated admin frontend, from powerful templates to fast rendering, data management, HTTP services, form handling, and so much more. Last but not least, Angular presents us not only the tools but also design patterns to build our project in a maintainable way as it is strictly OOP and component based.

We choose to go with Angular for admin due to its dominated strength in enterprise application. Angular is well-known for its high performance, maturity and also its support for Typescript. On top of that, our developers have strong background in Angular for admin development and can quickly build the solution from the ground up.

Parse Server [3] is an open source version of the Parse backend (developed by Facebook) that can be deployed to any infrastructure that can run Node.js [4] which is a runtime environment for server-side JavaScript application. Its main strength lies in the ability to be self-hosted on any cloud computing platforms such as AWS or Digital Ocean and out-of-the-box features that simplify manipulating data client-side and server-side from MongoDB.

Figure 14: Backend Stack

Our client specifically wants Parse Server mounted on Node.js server using MongoDB for backend solution due to the superior flexibility, customizability and performance of this combination. Moreover, our development team also has proven experience and success in implementing this set of technologies, which inspires confidence of the decision.

1.1. Languages and Libraries

In order to achieve effective outcome, several languages and libraries [8] were used as following:

Figure 15: HTML Logo

- **HTML:** Hypertext Markup Language, is a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages. We choose it because it is fundamental web page building script.

Figure 16: CSS Logo

- **CSS**: is a simple design language intended to simplify the process of making web page presentable by define how to display HTML elements. We use it to style and make our website responsive.

Figure 17: Typescript Logo

- **TypeScript**: is a language for application-scale JavaScript. We use it for easier debugging and type checking which makes development faster.

Figure 18: Bootstrap Logo

- **Bootstrap:** is a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development that uses HTML, CSS and Javascript. We use it for mobile responsiveness.

1.2. Tools

The following tools help me in the project development in an effective and smooth manner:

Figure 19: VSCode Logo

- **Visual Studio Code:** is a code editor redefined and optimized for building and debugging modern web apps and cloud applications. We use this for smooth code writing and debugging.

Figure 20: Navicat Logo

- **Navicat for MongoDB:** gives a highly effective GUI interface for MongoDB database management which we use to create and manage MongoDB for our application.

Figure 21: Postman Logo

- **Postman:** is the essential toolchain which we use to share, test, document and monitor APIs of Parse Server.

Figure 22: Gitlab Logo

- **Gitlab:** is a web-based repository manager that lets teams collaborate on code, duplicate code to safely create and edit new projects, then merge finished code into existing projects. We use this for our internal version control.

Figure 23: Chrome Logo

- **Google Chrome:** is a powerful web browser that we choose for amazing browsing speed and contains concrete developer tool is great for debugging code.

2. Architecture of the Application

2.1. Physical Architecture

The physical architecture of the web application below illustrates the general concept of the web application that explains how the web application functions and the physical components of the web application as below.

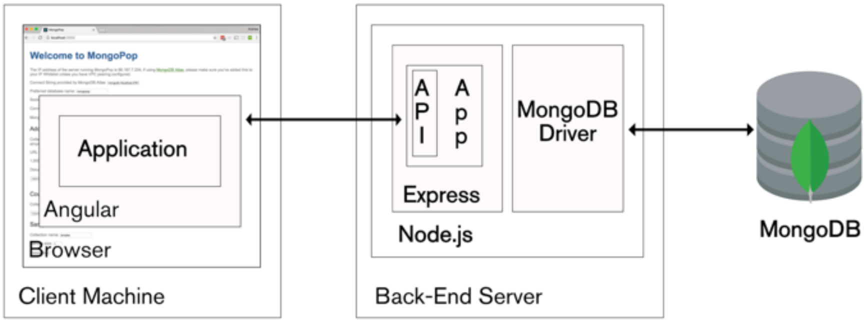


Figure 24: Physical Architecture

Based on figure 24, there are 3 major components of our physical architecture which are:

- Client Machine: represents the client web browser that retrieves and renders client and admin apps such as Nuxt and Angular with the data from Node.js server.
- Back-End Server: represents the Node.js server that serves both client apps including Angular and Nuxt combining with Parse Server Express app that is responsible for storing API endpoints and Cloud functions necessary for client data.
- MongoDB: represents the data storage server that communicates with Node.js server through the driver inside the backend.

2.2. Logical Architecture

In building this project, we combine client, server and admin on a single Node.js server to save resource and cleanly organize codebase in a mimicked MVC architecture [6]. Client and admin architectures which use Nuxt and Angular respectively are fundamentally component-based architecture. Parse Server, which contains our endpoints, are customized into a structure that contains endpoints in controller. Then, there is a shared directory that represents essential data models that simplify business logic with data mapping and accessing. Ultimately, after implementing the combination as mentioned, the Node.js infrastructure now is mimicked into a powerful MVC architecture that can be customized in any way the business logic requires. Of course, it is not a strict built-in MVC architecture like other frameworks such as Laravel due to the nature of customizability and flexibility of Node.js ecosystem. Below is an overview breakdown of the typical MVC based on figure 25 that is adhered in our system:

- Model: represents the shared data structures and normally contains accessors and customized functions for convenient data mapping on both ends and for interacting with Parse Server to retrieve objects' information.
- View: represent pages that are presented to user after data is provided and manipulated by the component methods or functions in controller.
- Controller: represent Parse Server endpoints and all the component methods that query and manipulate data on client side to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

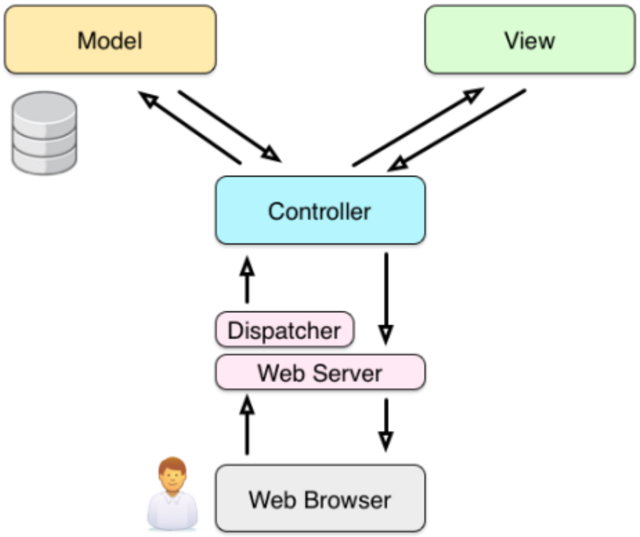


Figure 25: Logical Architecture

V. IMPLEMENTATION

1. Setting Up Environment

Based on figure 25, first of all we need to install Node.js and Express.js that will handles our client, server and admin on a single infrastructure.

Secondly, we have to install Angular via Angular CLI, and then Nuxt in universal mode which supports both server-side and client-side rendering. Then, install Parse SDK on both platforms to use Parse out-of-the-box features on client side.

Thirdly, Parse Server has to be installed as an Express app, configured database credentials to connect to MongoDB instance. Also, extract the app Id for later integration with Nuxt.

Forth, route Nuxt and Angular application to point to Parse Server using the app Id I mentioned earlier. Now both Nuxt or Angular are integrated with Parse Server and can be communicated with each other.

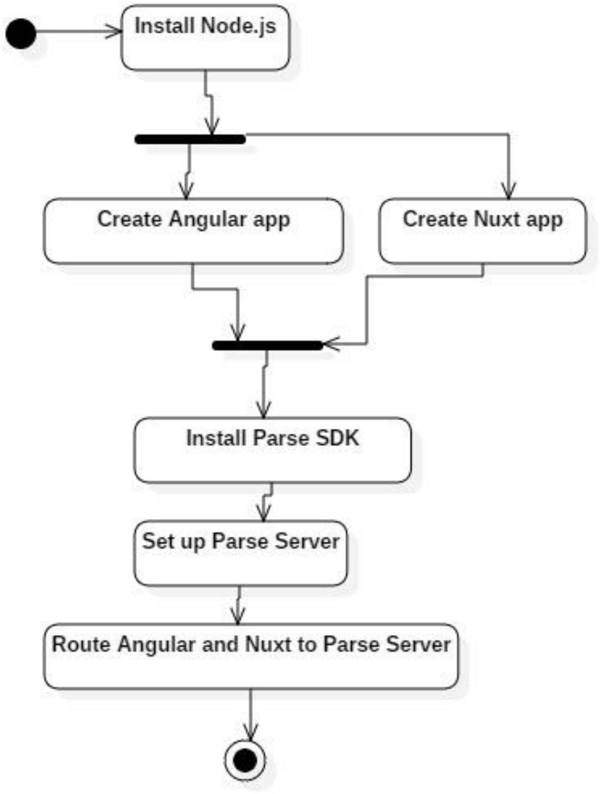


Figure 25: Environment Setup

2. Implementing Core Features

We have set up our environment in the most critical steps so now it is ready for us to implement our features on it. Since there are so many features and sub-features, I am going to focus on the perspective of the user flow from browsing the products to checking out and on the perspective of the admin flow performing CRUD operations.

2.1. Page Rendering

According to figure 26, when user visit each page, the **asyncData** method will be called with **ParseQuery** to fetch public page data from server before initiating server-side rendering and initiating components. The contents that are rendered server-side are public and can be viewed by visitors:

- Banner and Promotional Products
- Spotlight Products
- Product Detail
- Shop

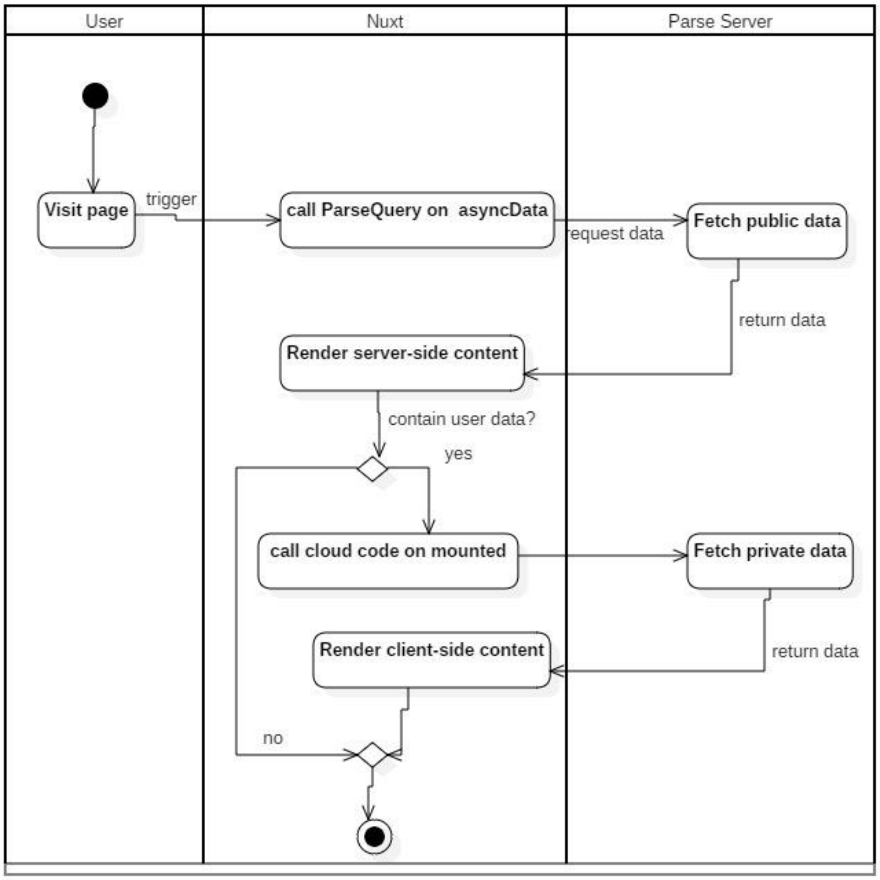


Figure 26: Nuxt Page Rendering

For sensitive user data such as cart, profile, orders, reviews and product personalization, we call cloud code on **mounted** method to query data from server and render it client-side This practice is for security purpose as we do not want to expose sensitive query functioin on client side.

2.2. Authentication

For activities such as ordering, reviewing or updating personal information, user has to authenticate to access such feature. User can either log in with email address or phone number through Firebase. Only three fields are required for registration which are username, full name and password.

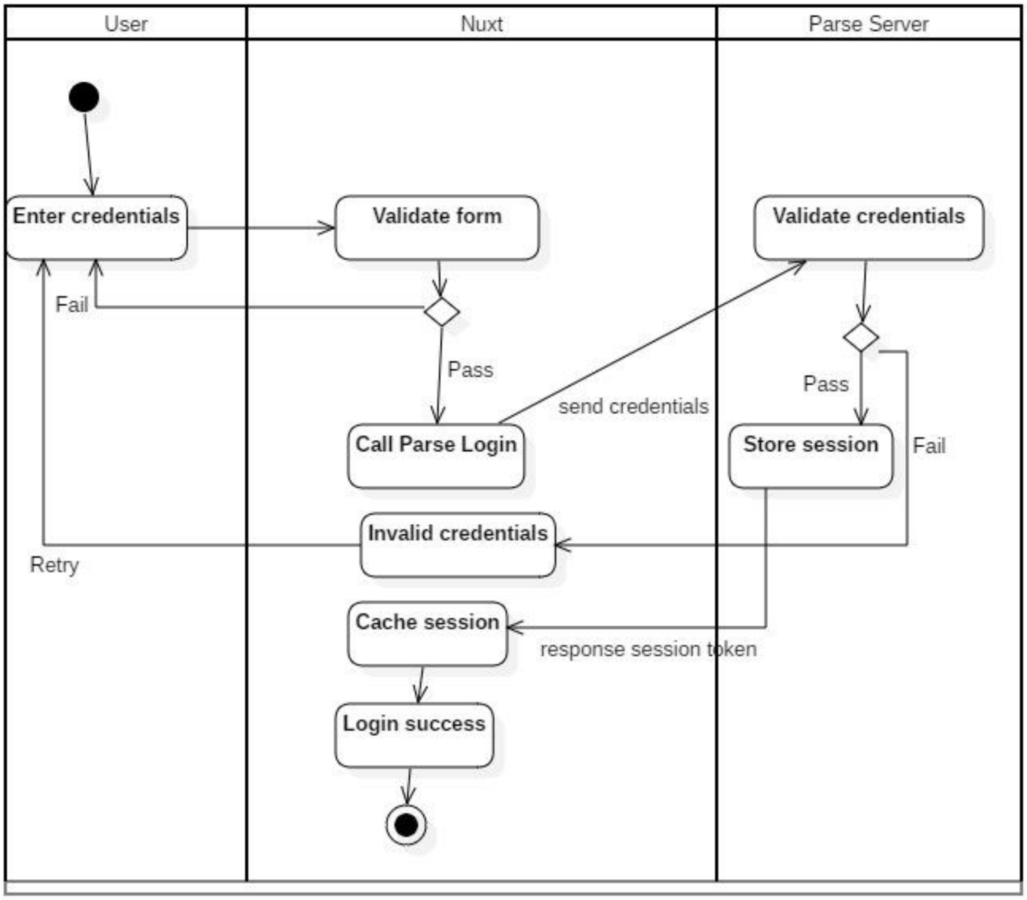


Figure 27: Authentication

According to figure 27, login is very straight forward as Parse SDK provides login method for that. First, user needs to provide credentials including username and password and then Nuxt will check if the data is valid on client-side. If the validation passes, Nuxt will simply call **Parse Login** method that contains credentials as parameters which automatically got send to server for validation. If the server validate the credentials successfully, it will renew or store session on server and response token to Nuxt. By default, Parse will cache the session data on client and the user is successfully.

2.3. Seeking the Right Product

User has so many ways to select the right product such as browsing collections from home page or just search for it. Whether they select a collection or search, user will be directed to shop page which has powerful filters to single out their desired product too.

According to figure 29, whenever user perform search or filter product, the relevant parameters will be extracted and push to URL. Nuxt application will listen to URL change and call Parse Query using the parameters to query product data from server through cloud search which will return the product results back upon completion.

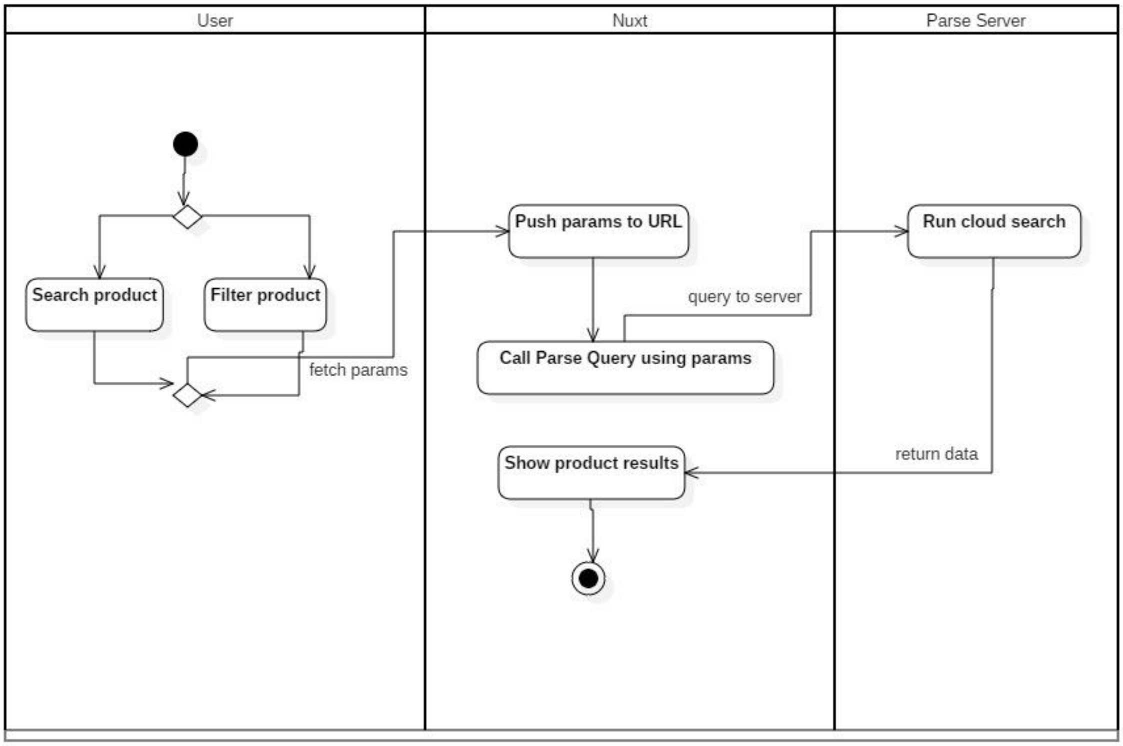


Figure 29: Seeking Product

Having clicked a collection or searched for a product, user is directed to product result page. We provide many filters such as rating, category, brand, top sale and popularity and even price range. Selecting any of these filters will trigger product fetch method then dynamically and instantly update the page with new product results thank to the power of Nuxt client-side rendering combined with Parse Query. There is no need to reload and the update is done almost real-time which keeps the user experience at the very excellent.

After filtering for their desired item, user can now select their favorite product and will be directed to product detail page to find out the information and review.

According to figure 30, when user has clicked on a product, it will also trigger a cloud function to record product view of according user which is important data for getting popular products and the Just for You method that suggests relevant products to user. Products from top five categories that user has clicked on the most will be suggested on homepage.

On the product detail page, product has color, option, number of stock available, description and summary of past ratings and reviews to help user decide as well as related and recently viewed products generated by personalization cloud code function I mentioned earlier when user clicks on certain product. These data are of course queried on the post through Parse SDK.

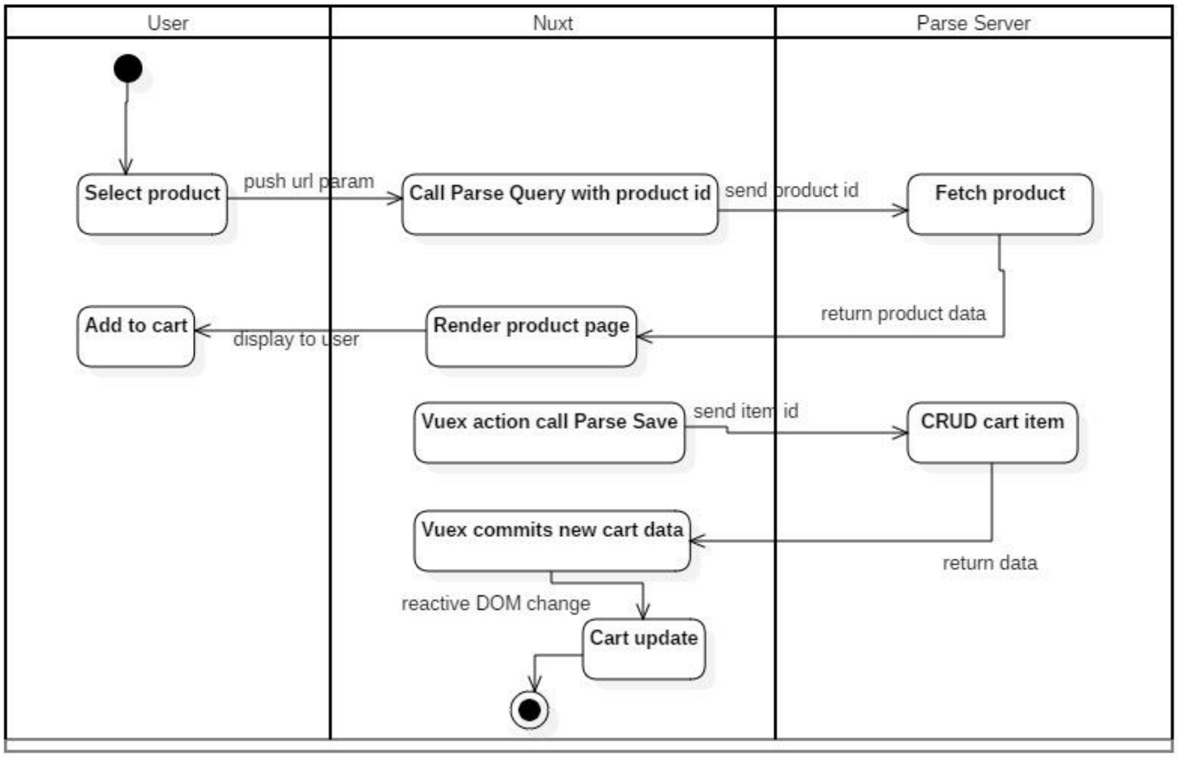


Figure 30: View Product

User now can decide to set their purchase quantity and buy now or add to cart. If user add to cart, it will trigger function to save the item to cart in Vuex store and then user can shop and add more as they like. User can also delete product from cart from the upper right cart section and Vuex will updates the cart accordingly. After user has decided to buy now or click checkout their cart items, user will be redirected to checkout page.

2.4. Checkout and Payment

We try to implement the checkout process as simple as possible and thank to Vuex store and PayPal Smart Payment Buttons and their API we successfully manage to give a seamless checkout experience to user.

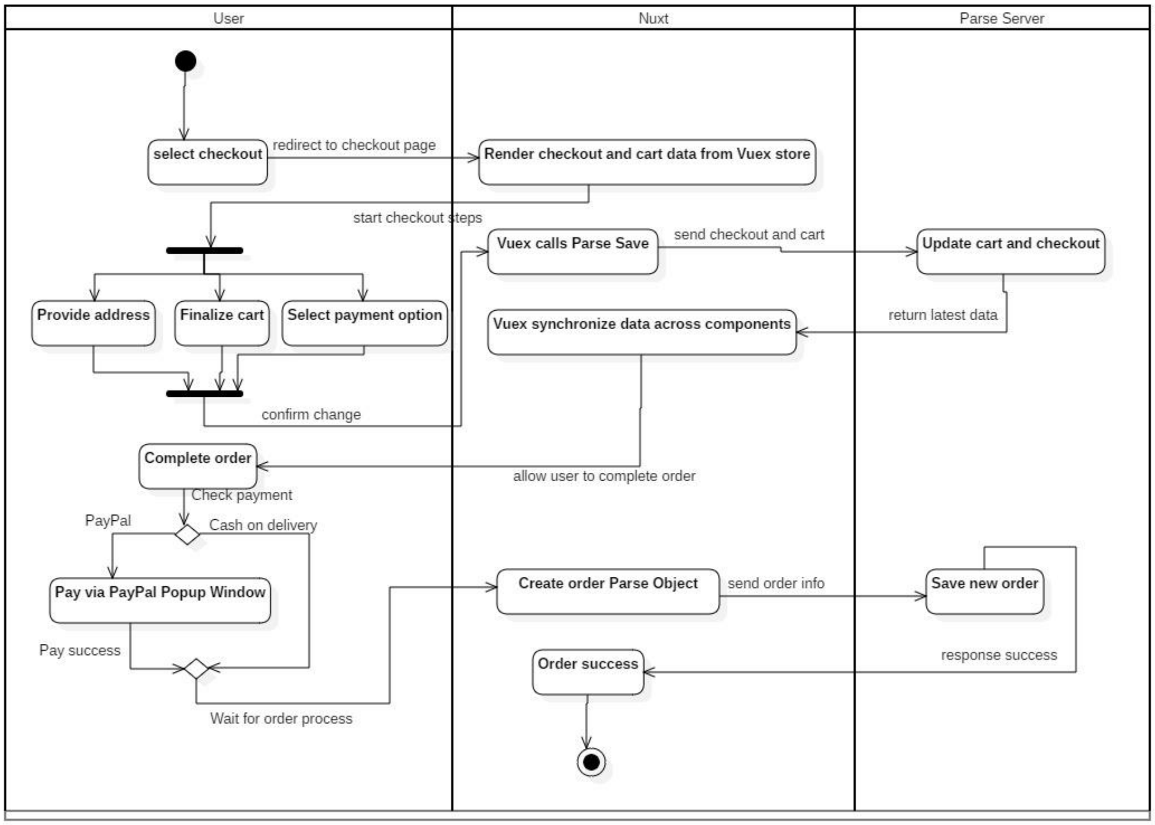


Figure 31: Payment and Checkout

According to figure 31, first, user has to select the items to checkout and provide their shipping address. This is unnecessary if they already have provided the default address in their profile but they can still modify or add new in the checkout page. The address must include phone number or email for contact and their shipping address which includes detailed address plus their city or province. Having provided these info, it will be updated to checkout object in Vuex store for payment step.

For the payment step, if the user choose payment on delivery, the user can complete the checkout process immediately and then the user will be directed to successful checkout page and greeted with thank you message. Else, if the user selects cashless payment (Mastercard/Visa), then there is another crucial step to complete which is the PayPal Check Smart Payment Buttons [7]. According to Figure, after the cashless payment option is selected and the user select complete order button, the complete order button will be changed to PayPal Smart Payment Button and be put in loading stage with callback on an interval to listen and wait for successful PayPal payment from PayPal order API. The PayPal payment page will pop up immediately in another window for user to complete the payment. Upon payment completion, PayPal Order API will return success response to the button callback with transaction info and the checkout process will be completed and redirected to order success page. To explain the process in high level concept, here is big-picture view of how Smart Payment Buttons integration based on figure 32 works:

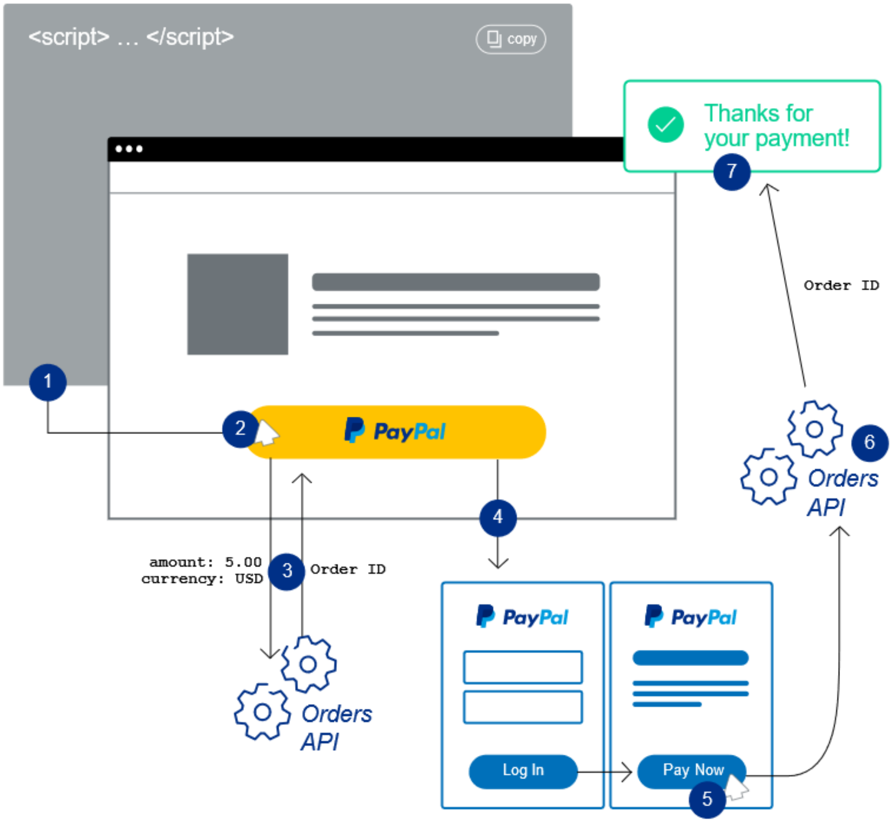


Figure 32: PayPal Payment

1. Show the PayPal Smart Payment Buttons on the checkout page.
2. Buyer clicks the button.
3. The button calls PayPal Orders API to set up a transaction.
4. The button launches the PayPal Checkout experience.
5. The buyer completes the payment.
6. The button calls PayPal Orders API to finalize the transaction.
7. Show a confirmation to your buyer and redirect to order success page.

On the order success page, the user can now click to return homepage and start browsing for more products and add to cart and the cycle continues. But where does the order go? The user has to have a way to view the successful order and track the delivery status to get their items.

2.5. Tracking Order

According to figure 33, order can be simply tracked by going to order list and then click on track order. According to figure, when the user select on track order, Nuxt application will call Parse Query to fetch the latest order information from server and then the data will be displayed back to user on the application.

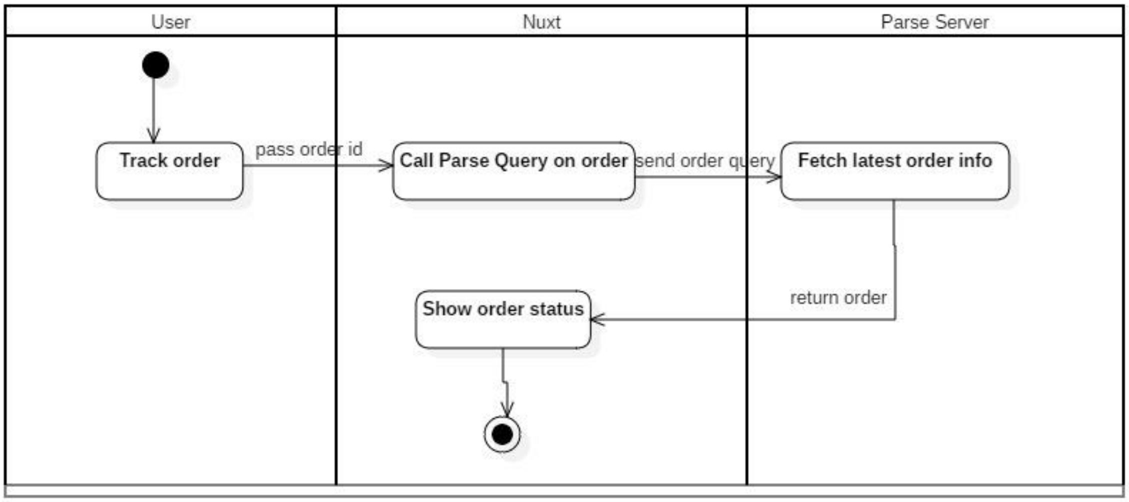


Figure 33: Track Order

2.6. Reviewing Product

Reviewing product is simple – according to figure 34, user just click the review button and get redirected to the review page that contains all the product reviews fetched by **Parse Query**. Each review is linked to product variant that the user order and only require rating and the review detail is optional. User doesn't have to review all the products they purchase as it is much more simple and convenient for them to come back later to complete the remaining reviews. When the review is done by at least giving a rating and selecting the submit button, **Parse Save** will be called on the review in order to save it to server and the product review field get dynamically updated to contain thank you message and the picture of the product.

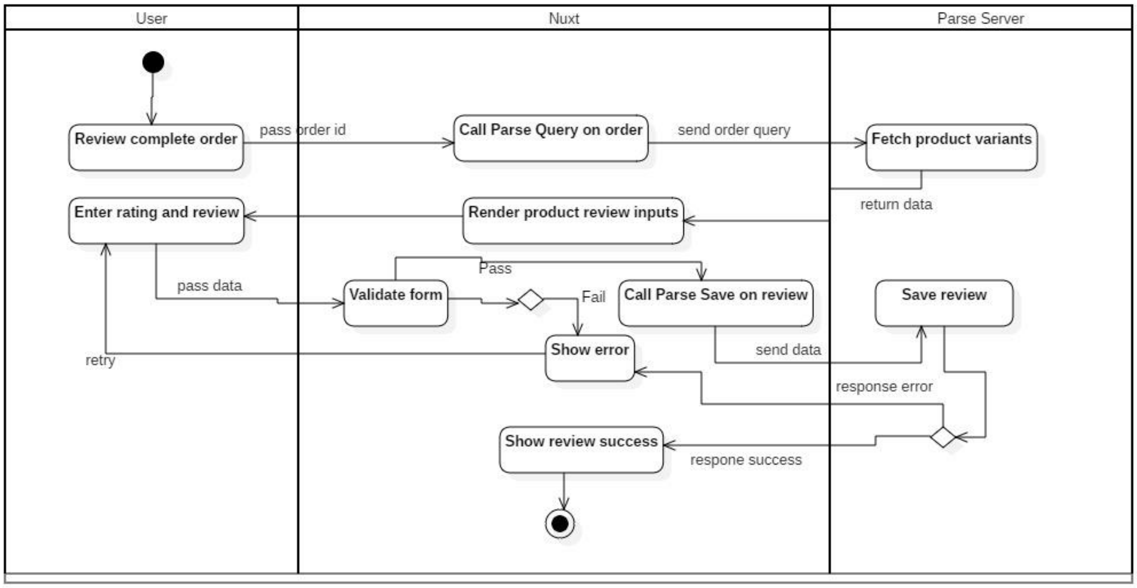


Figure 34: Review Product

2.7. Managing Profile Information

According to figure 35, when the user update profile, the information will be first validated on client side and if it passes, it will trigger Vuex method that call **Parse Save** on the user object to server and after the server has returned the latest data, Vuex will synchronize the change across all related components that use user object and the profile is successfully updated now.

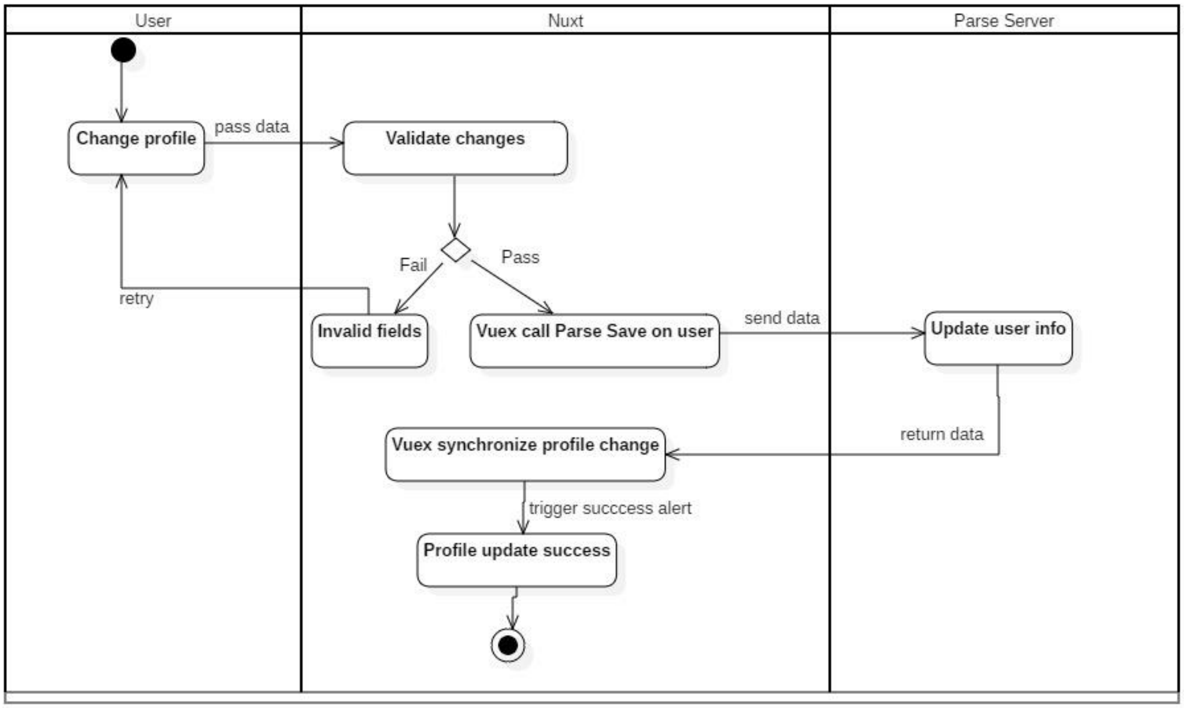


Figure 35: Change Profile

2.8. Managing and Tracking Users

For business or security reason, admin may need data related to product view, hot search keywords and user activities such as when user authenticate or make an order. According to figure 36, to make this possible, we implement API for specific tracking feature that automatically triggers and save every user actions such as authentication, search and product view. Each user activity will trigger call to monitoring API that will record the action and relevant data.

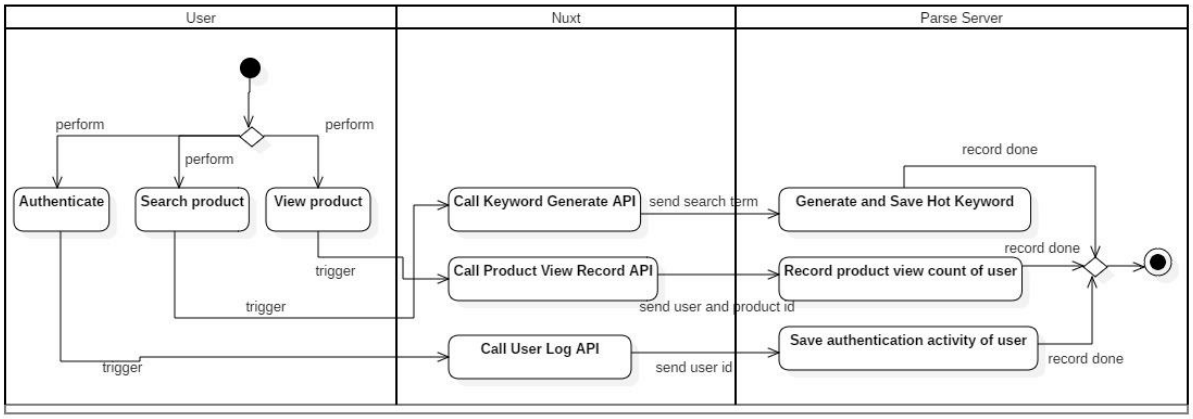


Figure 36: Track User

Regarding to permission to track issue, we do include this in part of term of condition which the user has to accept in order to register for an account. This is necessary as we need the data to personalize user experience as well as improving ecommerce business to serve users better.

2.9. Managing User's Orders

Admin can see all the orders made in order list which contains the name of user who made the order, the order total cost, the status and the order date. Admin can filter out by date range, type of order status, username and search and the interface will dynamically update by calling Parse Query with the necessary parameters to fetch the latest data without inconveniently reloading the page.

According to figure 37, when the user has completed an order, admin can either accept, complete or reject the order if necessary. Admin just needs to click on a specific order and be directed to order detail page that contains summary of the order including the shipping information and products purchased as well. For new order, if the admin choose to accept the order, the order status will be update to accepted status and order status is displayed to user in frontend accordingly. Then, when the order has physically been delivered and paid by the user who's made that order, admin can choose to complete the order and the process is done and user can make review on frontend. On the other hand, if the admin just straights up reject new order, the order will be marked as rejected and displayed in rejected order section in user frontend.

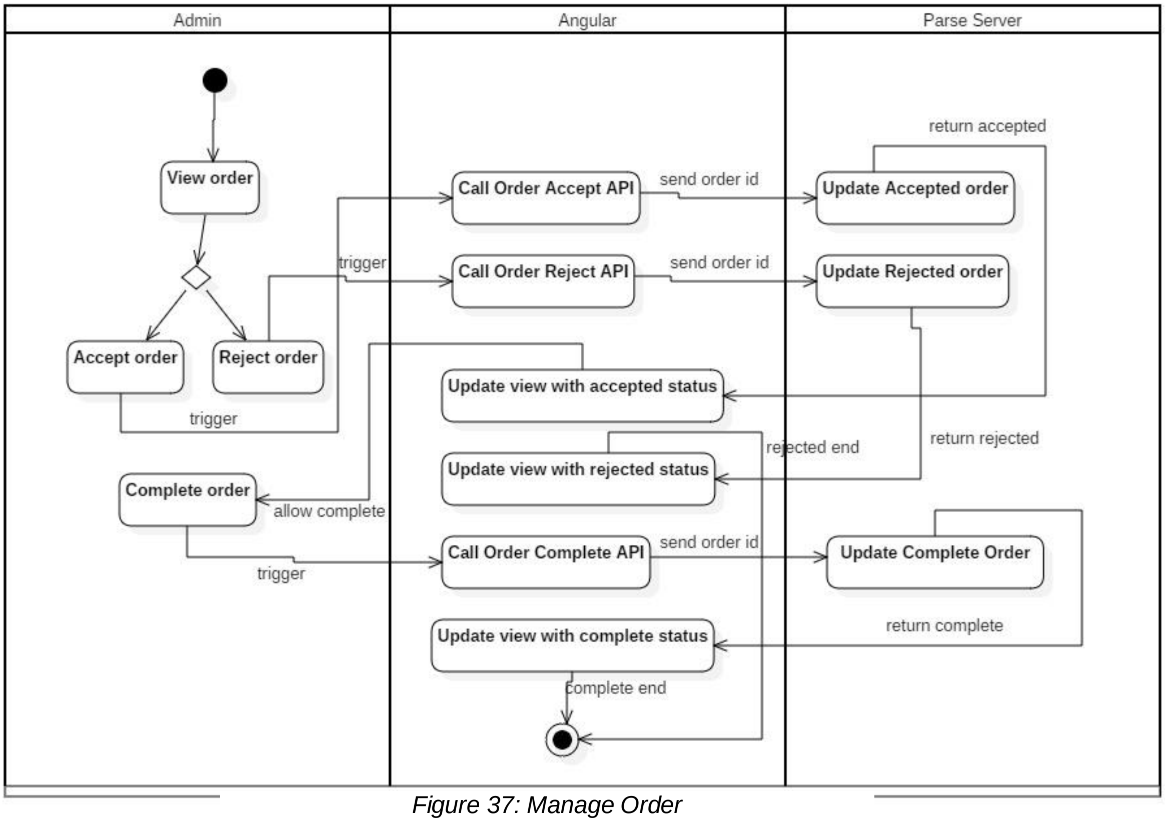


Figure 37: Manage Order

2.10. Content CRUD Operations

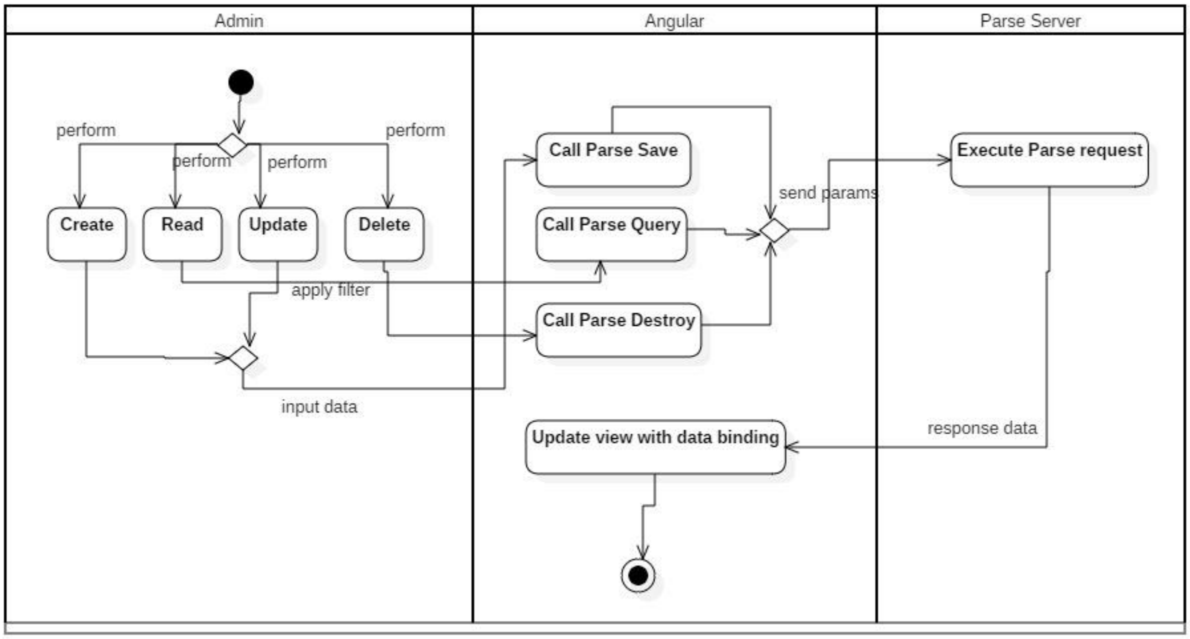


Figure 38: CRUD Operations

As I have mentioned that all the content in frontend are dynamic, admin side is where we manage all of them. Those include **promotional banners, product categories, spotlight, product collection, shipping zones, static pages, stock, meta tags, configurations footer menus** and most important the **product list** itself.

According to figure 38, these contents follow similar formula to perform CRUD on them and the difference lies in the unique properties of each content. **Parse Save** is for creating and updating object**. Parse Query** is for reading information and **Parse Destroy** is for deleting the object. They also use various filters that vary by content such as search on title, range of created dates, ownership and type of content. Content can also be sorted by order manually to achieve the most relevant user experience in frontend. Upon data response, the view will be bound to the data and rendered dynamically.

VI. CONCLUSION

1. Results

After four months of effort, PSANAT ecommerce project is 95% complete with the remaining tasks left are checkout with another cashless provider, testing, UI bug-fixing and planning future feature. I did my best and manage to contribute approximately 50% of the whole project on both client and admin side.

I complete all the important functions ranging from assigned to me which as below:

Functionality	Status
• Checkout with both PayPal and Cash on Delivery • Manage and track order • User personalized products • Log user's activities and product view • Product search, filter, listing and view detail • Hot search keyword • Manage profile information • Subscription • Sale Report Dashboard • Top sold and viewed products • Manage user and user's activities • Manage page • Manage order • Manage hot keyword • Manage configuration • Export to excel	**Complete**
• Checkout with ABA Pay • Store Cashless Credentials on site	**In Progress**

Table 7: Result

The features I have completed on client side smoothly help user to seek the desired product and checkout their order seamlessly with both cashless and cash option. Furthermore, I successfully provide a way for user to track order and manage their profile information at ease. The recommendation and hot search features assists user in decision of purchasing relevant and trendy products tailed for them. On the other hand, on the admin side, admin can conveniently manage order and user as well as be able to track their sale performance along top products.

The remaining functionality is checkout with ABA PAY. Apart from that, the project is still in testing and debugging stage and possible new feature will be proposed and added soon.

2. Strong Points and Weak Points of the Web Application

No solution is perfect, so same goes for my current web solution. It is uniquely designed and developed to fit specific purpose so it has its strong points and weak points.

Strong Points	• User can easily seek product and checkout instantly in a few clicks thank to fast performance from Nuxt. • It has great SEO support due to the combination of both server-side and client-side rendering based on universal mode of Nuxt. • Super intuitive and user-friendly interface making user be able to navigate across the website smoothly without any hassling reload. • Dominant performance and ease of maintainability due to the usage of Typescript across client, server and admin project. • Building on top of Node.js makes the system very scalable and customizable in the future as the platform is not locked to a single technology but instead supports multiple applications to save resource and provide clean centralized architecture.
Weak Points	• Interface Bugs related to responsiveness of different screen sizes. • Product Recommendation uses normal algorithm and it is not efficient for large data. • Code needs to be refactored to clean and optimize.

Table 8: Strong and Weak Points

3. Difficulties

Life is a struggle and full of difficulties, but that only makes us stronger. Throughout my period of the internship, I faced several challenges both technically and personally:

When I first joined Coolbeans, I severely lacked in Node.js which is the main technology used there. I managed to overcome this obstacle by start learning and asking from both Google and my senior, and eventually I got very proficient at it and I can safely say I am proud of my knowledge and skill in the Node.js ecosystem now.

I have been struggling financially to support myself to be independent. Most of the time, I feel like giving up but the goal of becoming a superior software and machine learning engineer keeps me going. There are just so much more to learn and improve on myself, so I try to pick myself up by visualize the ultimate success and seek mental support from my significant other regularly to keep in order to get to that goal. Its takes time but I am on my way indeed.

4. Experiences

I used to intern at Smart company before and I gained a lot of professional and technical experience from there. This time, at Coolbeans, I applied those experience and gained even more. The complexity of PSANAT ecommerce project has taught me so much about good code practices and collaboration with both designer and developer. Unlike standalone project I used to do at Smart, every function and view written has to be reusable later in other parts of the system to keep it clean and easier for other developers to read. When there is need to develop new feature, I have to consult with designer to find the best approach first. If it is complicated to develop directly, designer will take responsibility of designing the feature first and assign to me later. If it is simple, I can go ahead and develop on my own but later I have to get feedback and correct based on it. The lesson here is to do what I do best and respect other team members' skillset and have trust in them. In term of work environment, Coolbeans is much more relaxing and flexible than company like Smart because it is consisted of a small team. The most benefit of this is I can learn directly from senior who works in the same room and can communicate and comment on my work frequently which is great for my self-development. It teaches me to love constructive criticism and not take things personal because criticizing is focused on the code not the person. Last but not least, besides work, our team has fun chitchatting about nerd stuffs which is useful for teamwork spirit and we often have team celebration in case there is birthday or exciting landmark.

5. Perspective

Like I mentioned in weak points, the product recommendation needs rework based on machine learning. We also have eyes on future features including checkout with ABA PAY and possibly integrating machine learning in other parts of the ecommerce site such as search and related product. Moreover, it is getting an official release in the next two or three months and after that native mobile version will be planned.

6. Conclusion

I have no doubt the complete PSANAT ecommerce website is going to launch smoothly and attract a lot of users due to its performance and flexibility thank to the parallel combination of Nuxt, Angular and Parse Server on a single Node.js infrastructure. Big thanks to Coolbeans who has contributed a lot to my growth both professionally and technically. I have nothing but marvelous gratitude to my supportive team for the last four months.

Looking back, it has been a great journey studying at NIPTICT until working here. Technically, with the experience from both Smart and Coolbeans, I have become a capable junior developer in both Node.js and Laravel world. It makes me think and reflect on the people and experience I have met that make me who I am today. I owe it to NIPTICT and all the amazing and supportive lecturers who have commuted so far to teach me and other students everyday tirelessly. Maybe not all the knowledge got in my little brain due to my laziness and carelessness back then, but it definitely acts as a great foundation and inspiration that jump-starts my passion in developer career. And of course, all the amazing friends or classmates I have made at NIPTICT have been the most fun and supportive people I have ever met. Apart from that, I have faced many significant experience from NIPTICT – first true reciprocal love I wish I had tried harder in, nervous but rewarding entrepreneurship journey of SmartStart and extroverted communal living at dormitory. They are all amazing experience that changed my life and without NIPTICT I wouldn’t have experienced all those.

NIPTICT has achieved so much for a new university that has just been established for four years only. I praise NIPTICT’s continuous effort in advocating entrepreneurship, artificial intelligence and a digital Cambodia as a whole. I, a NIPTICT student, am proud to be a part of second generation students of NIPTICT, and I will keep fighting to advance myself further going forward and hopefully will be able to make contribution back to NIPTICT one day.

REFERENCES

[1]. Nuxt.js Documentation: *https://nuxtjs.org/guide*

[2]. Angular Documentation: *https://angular.io/docs*

[3]. Parse Server Documentation: *https://docs.parseplatform.org/js/guide/*

[4]. Node.js Documentation: *https://nodejs.org/en/docs/*

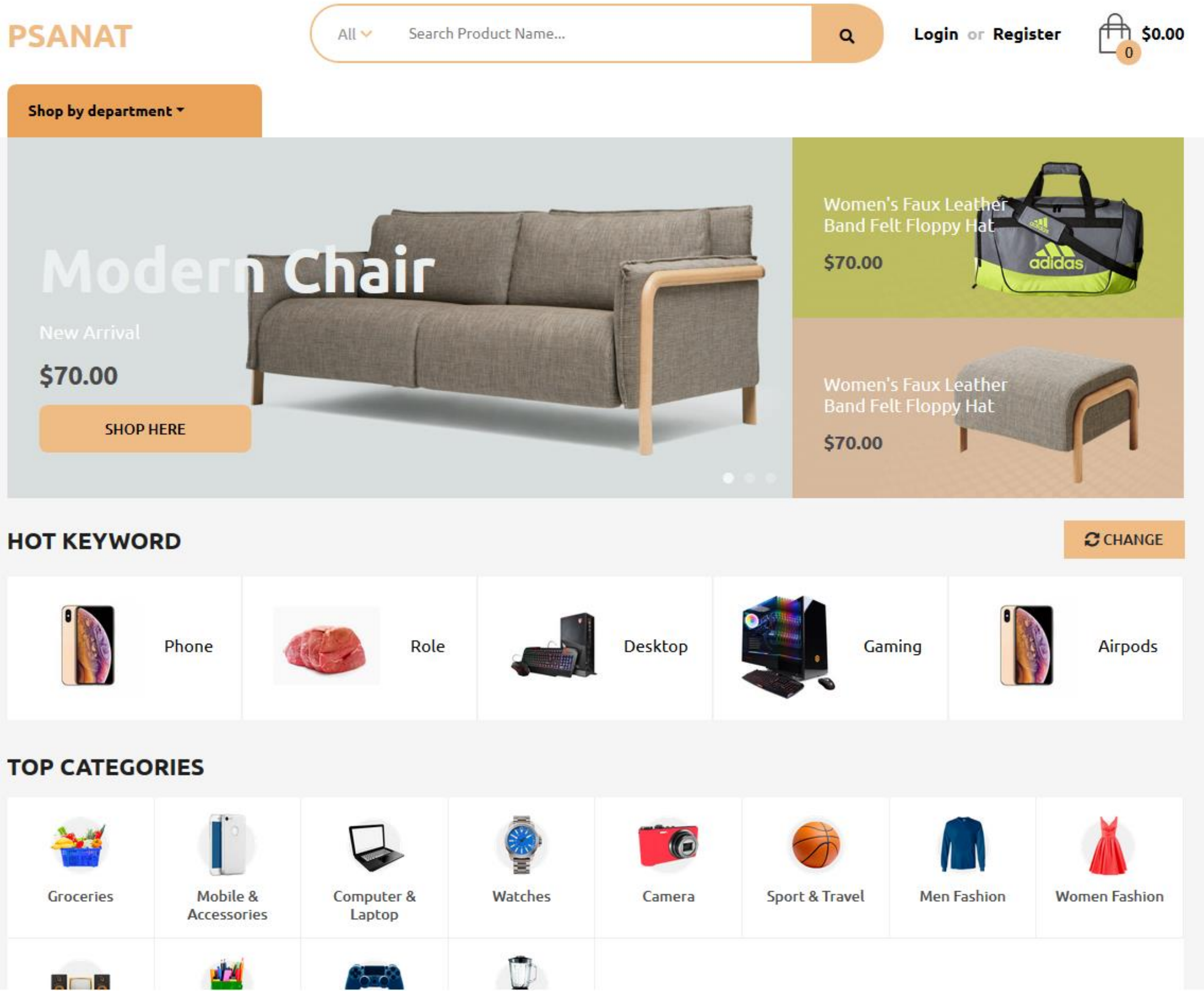
[5]. Coolbeans Official Website: *https://coolbeans.studio/*

[6]. The Modern Application Stack: *https://www.mongodb.com/blog/post/the-modern-application-stack-part-2-using-mongodb-with-nodejs*

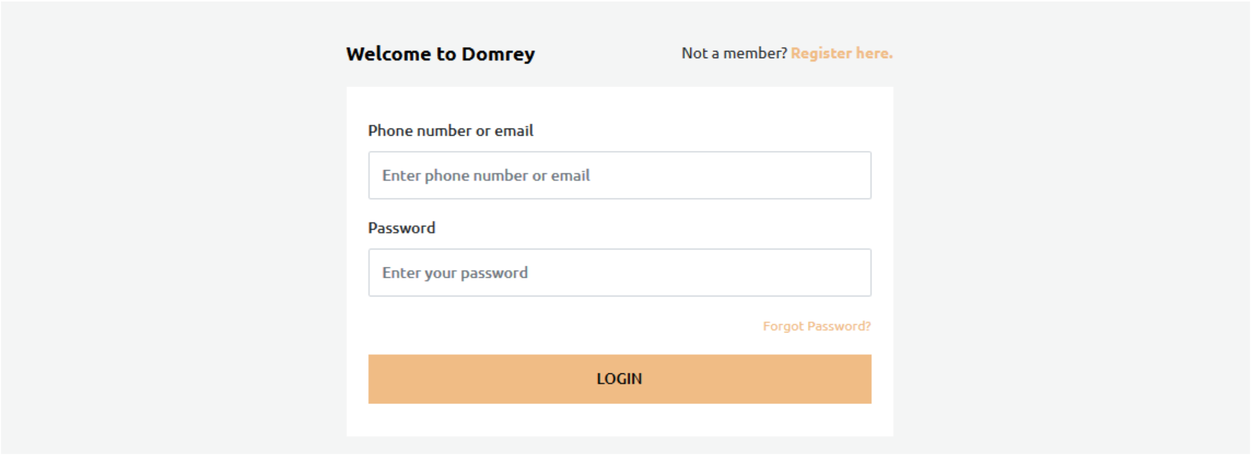
[7]. PayPal Smart Buttons Overview: *https://developer.paypal.com/docs/checkout/*

[8]. Web Libraries Resource: *https://www.tutorialspoint.com/tutorialslibrary.htm*

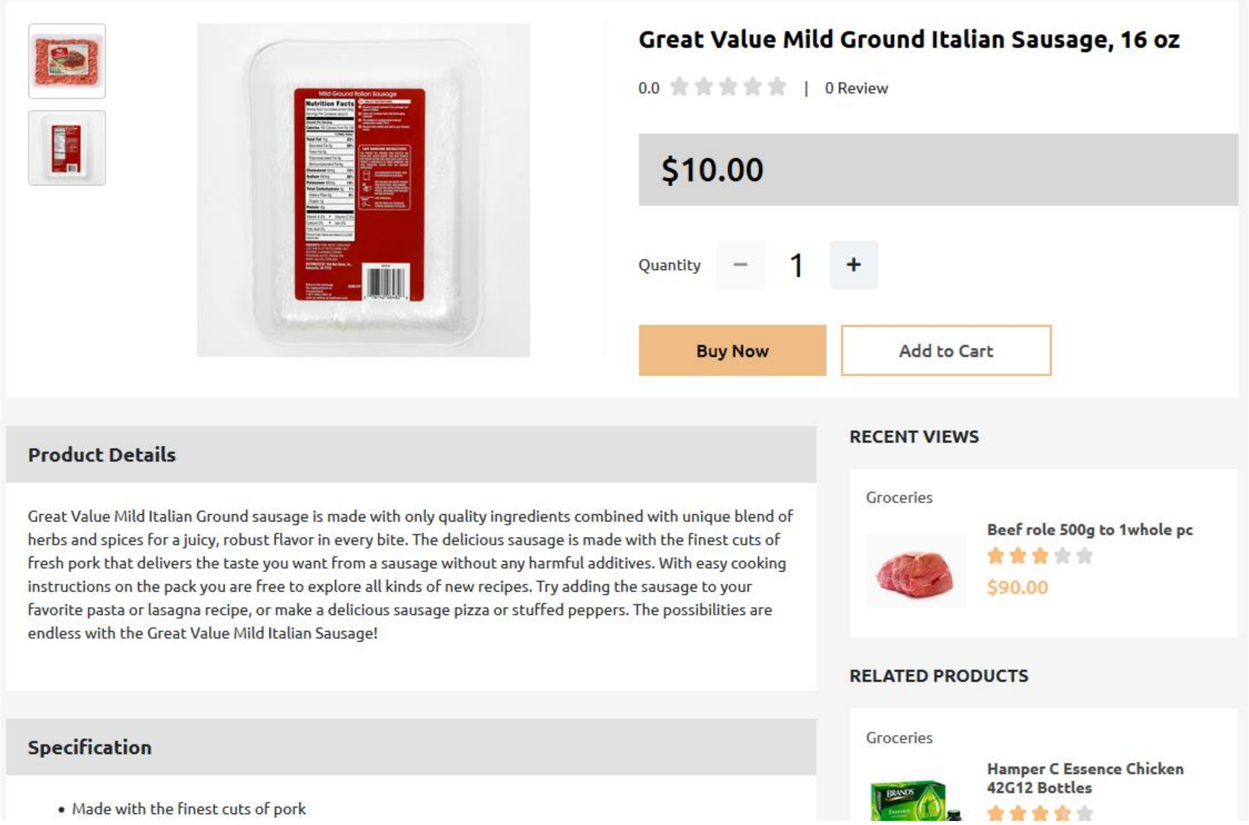
APPENDICES



Appendix 1: Homepage



Appendix 2: Login



Great Value Mild Ground Italian Sausage, 16 oz

0.0 ☆☆☆☆☆ | 0 Review

$10.00

Quantity: 1

Product Details

Great Value Mild Italian Ground sausage is made with only quality ingredients combined with unique blend of herbs and spices for a juicy, robust flavor in every bite. The delicious sausage is made with the finest cuts of fresh pork that delivers the taste you want from a sausage without any harmful additives. With easy cooking instructions on the pack you are free to explore all kinds of new recipes. Try adding the sausage to your favorite pasta or lasagna recipe, or make a delicious sausage pizza or stuffed peppers. The possibilities are endless with the Great Value Mild Italian Sausage!

Specification

- Made with the finest cuts of pork

RECENT VIEWS

Groceries

Beef role 500g to 1whole pc

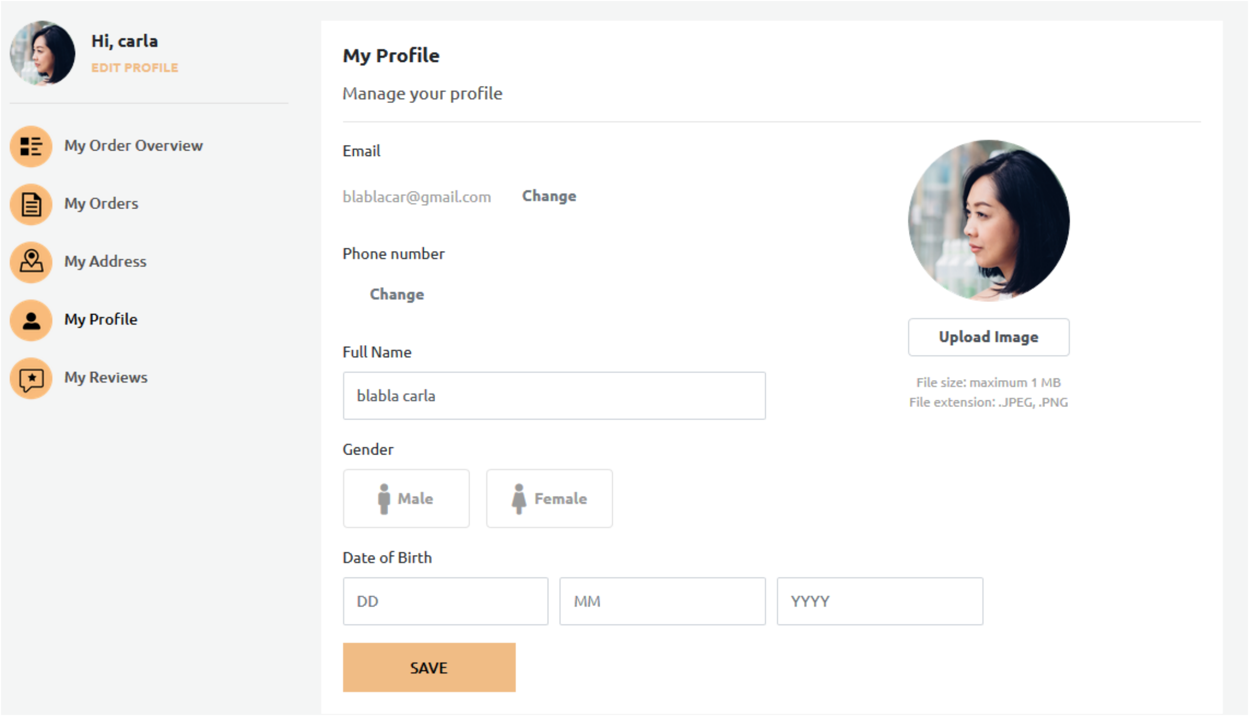
$90.00

RELATED PRODUCTS

Groceries

Hamper C Essence Chicken 42G12 Bottles

Appendix 3: Product Detail



Hi, carla

My Profile

Manage your profile

Email: blablacar@gmail.com

Phone number

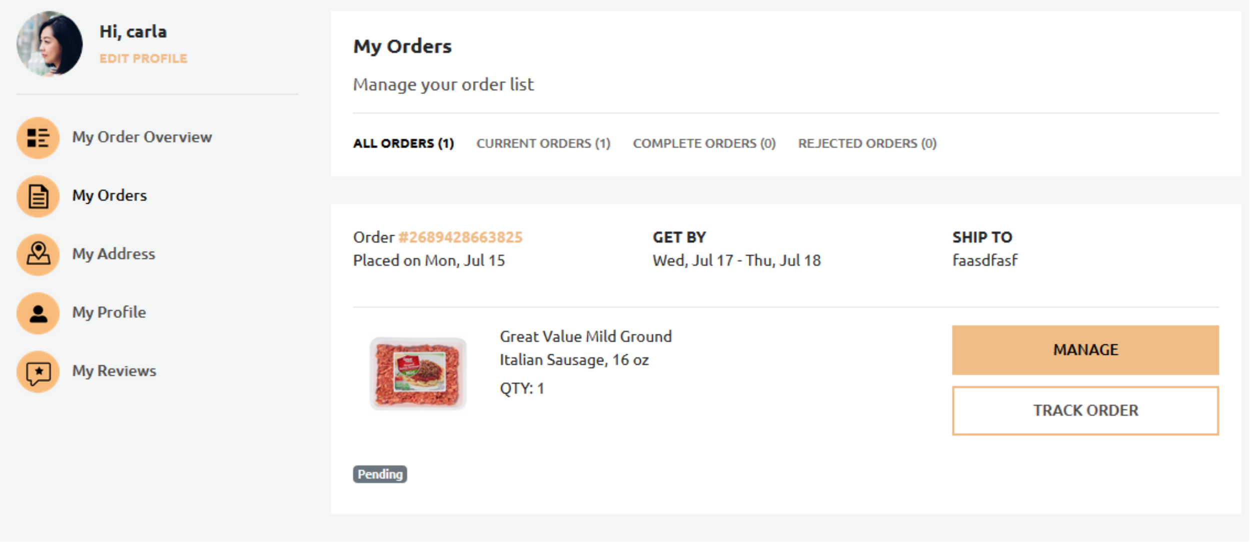
Full Name: blabla carla

Gender

Date of Birth: DD MM YYYY

File size: maximum 1 MB
File extension: .JPEG, .PNG

Appendix 4: User Profile



My Orders

Manage your order list

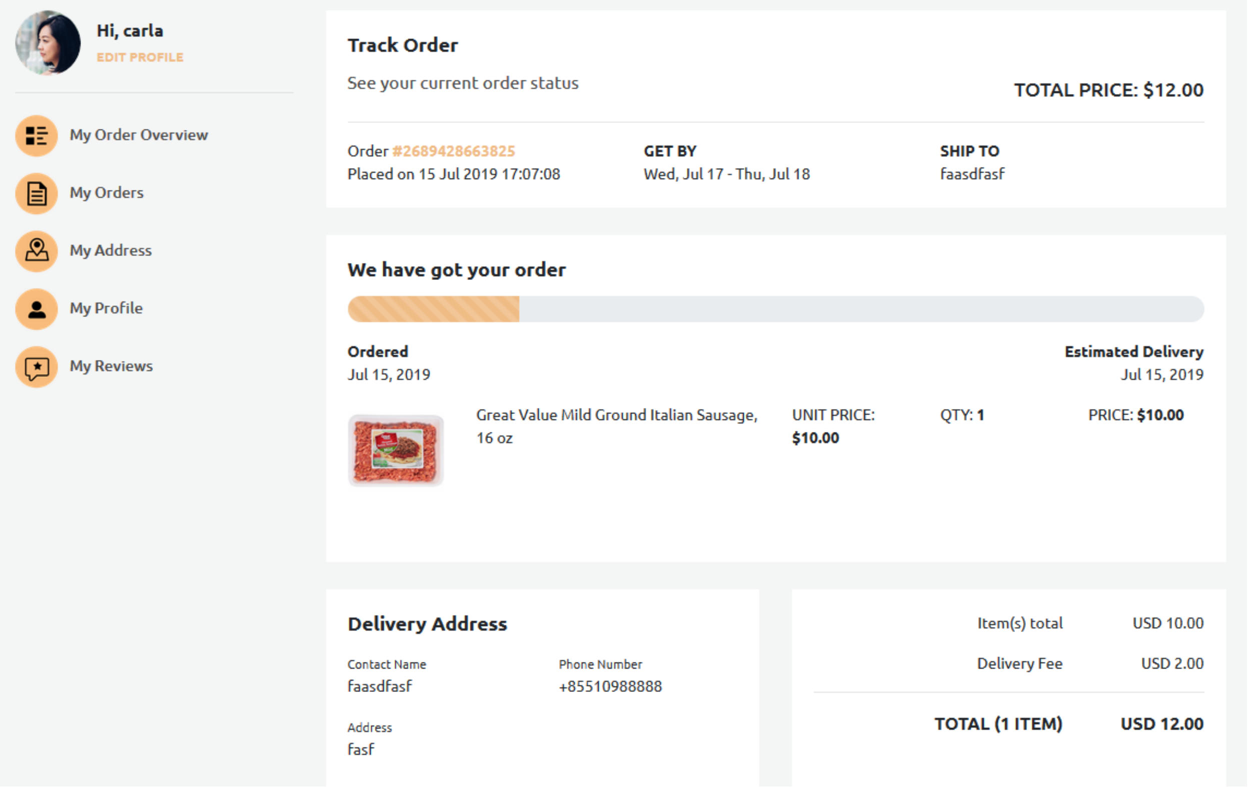
Order #2689428663825
Placed on Mon, Jul 15

GET BY
Wed, Jul 17 - Thu, Jul 18

SHIP TO
faasdfasf

Great Value Mild Ground Italian Sausage, 16 oz
QTY: 1

Appendix 5: My Orders



Track Order

See your current order status

TOTAL PRICE: $12.00

Order #2689428663825
Placed on 15 Jul 2019 17:07:08

GET BY
Wed, Jul 17 - Thu, Jul 18

SHIP TO
faasdfasf

We have got your order

Ordered
Jul 15, 2019

Estimated Delivery
Jul 15, 2019

Great Value Mild Ground Italian Sausage, 16 oz
UNIT PRICE: **$10.00**
QTY: **1**
PRICE: **$10.00**

Delivery Address

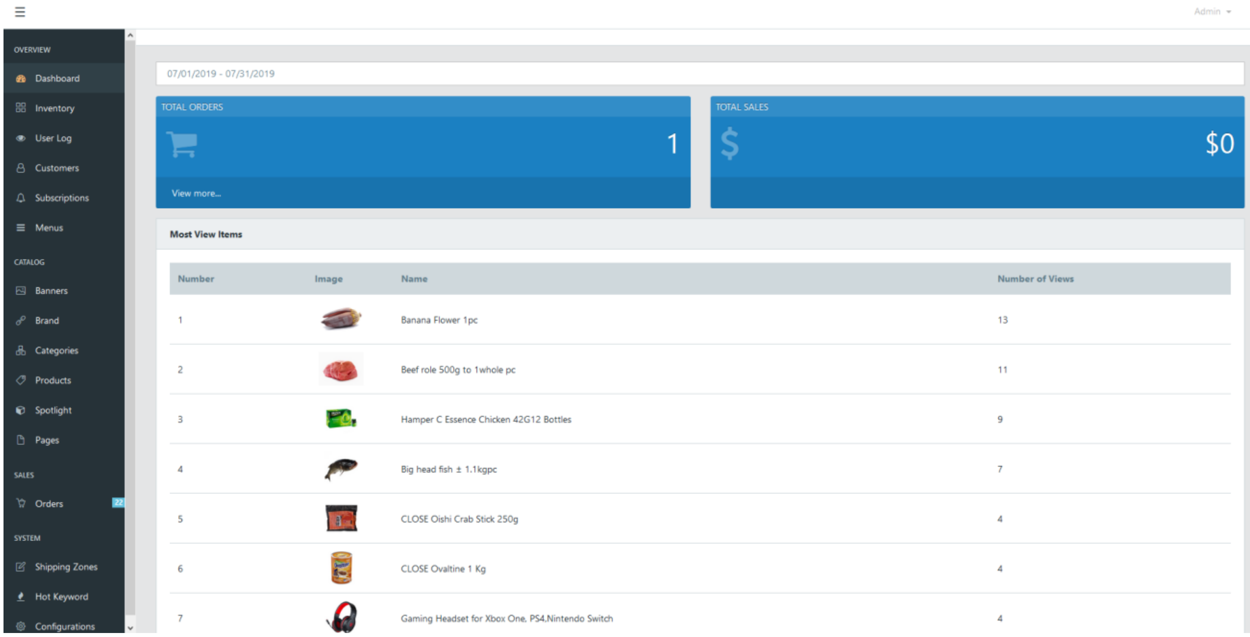
Contact Name
faasdfasf

Phone Number
+85510988888

Address
fasf

Item(s) total	USD 10.00
Delivery Fee	USD 2.00
TOTAL (1 ITEM)	**USD 12.00**

Appendix 6: Track Order



07/01/2019 - 07/31/2019

TOTAL ORDERS 1

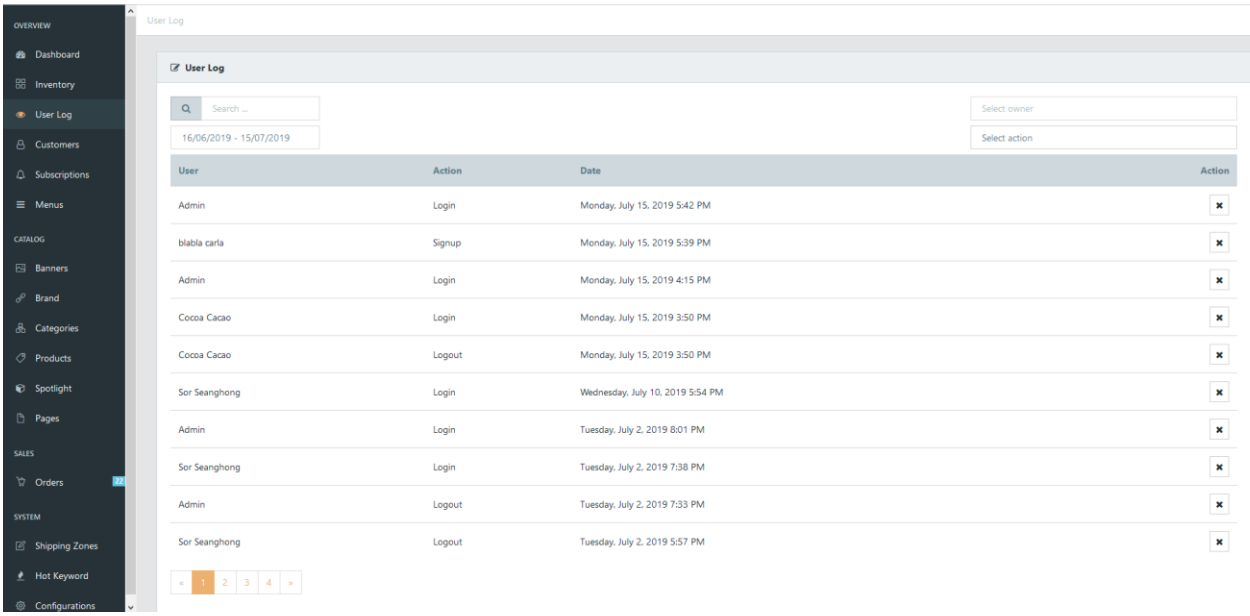
View more...

TOTAL SALES $0

Most View Items

Number	Image	Name	Number of Views
1		Banana Flower 1pc	13
2		Beef role 500g to 1whole pc	11
3		Hamper C Essence Chicken 42G12 Bottles	9
4		Big head fish ± 1.1kgpc	7
5		CLOSE Oishi Crab Stick 250g	4
6		CLOSE Ovaltine 1 Kg	4
7		Gaming Headset for Xbox One, PS4,Nintendo Switch	4

Appendix 7: Dashboard

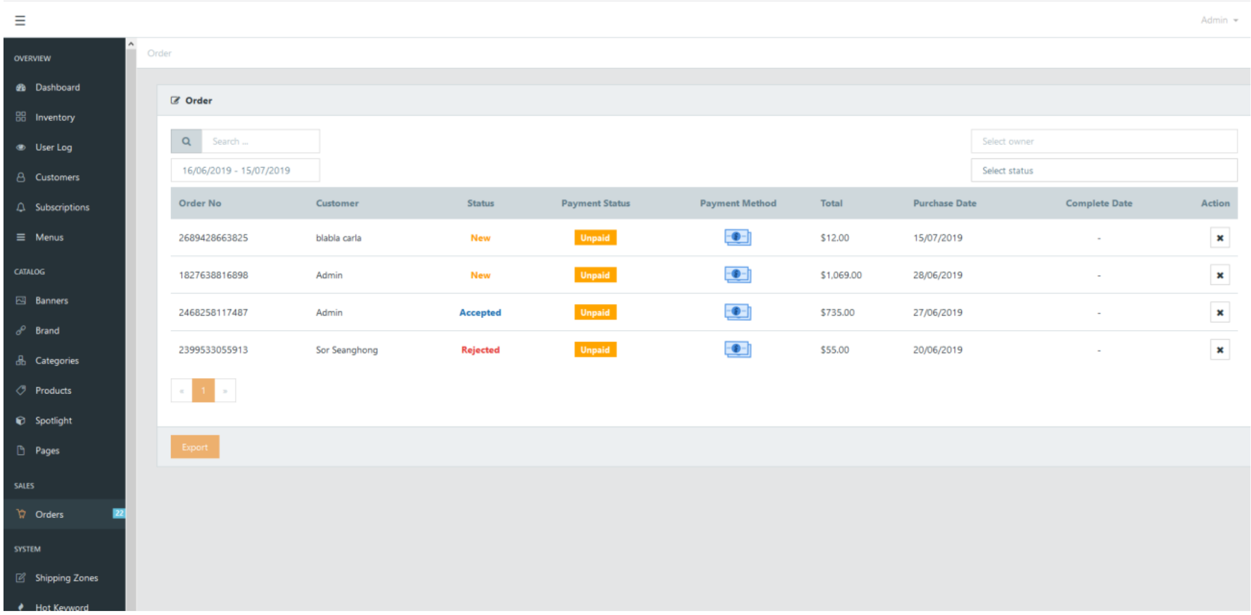


User Log

16/06/2019 - 15/07/2019

User	Action	Date	Action
Admin	Login	Monday, July 15, 2019 5:42 PM	✖
blabla carla	Signup	Monday, July 15, 2019 5:39 PM	✖
Admin	Login	Monday, July 15, 2019 4:15 PM	✖
Cocoa Cacao	Login	Monday, July 15, 2019 3:50 PM	✖
Cocoa Cacao	Logout	Monday, July 15, 2019 3:50 PM	✖
Sor Seanghong	Login	Wednesday, July 10, 2019 5:54 PM	✖
Admin	Login	Tuesday, July 2, 2019 8:01 PM	✖
Sor Seanghong	Login	Tuesday, July 2, 2019 7:38 PM	✖
Admin	Logout	Tuesday, July 2, 2019 7:33 PM	✖
Sor Seanghong	Logout	Tuesday, July 2, 2019 5:57 PM	✖

Appendix 8: User Log

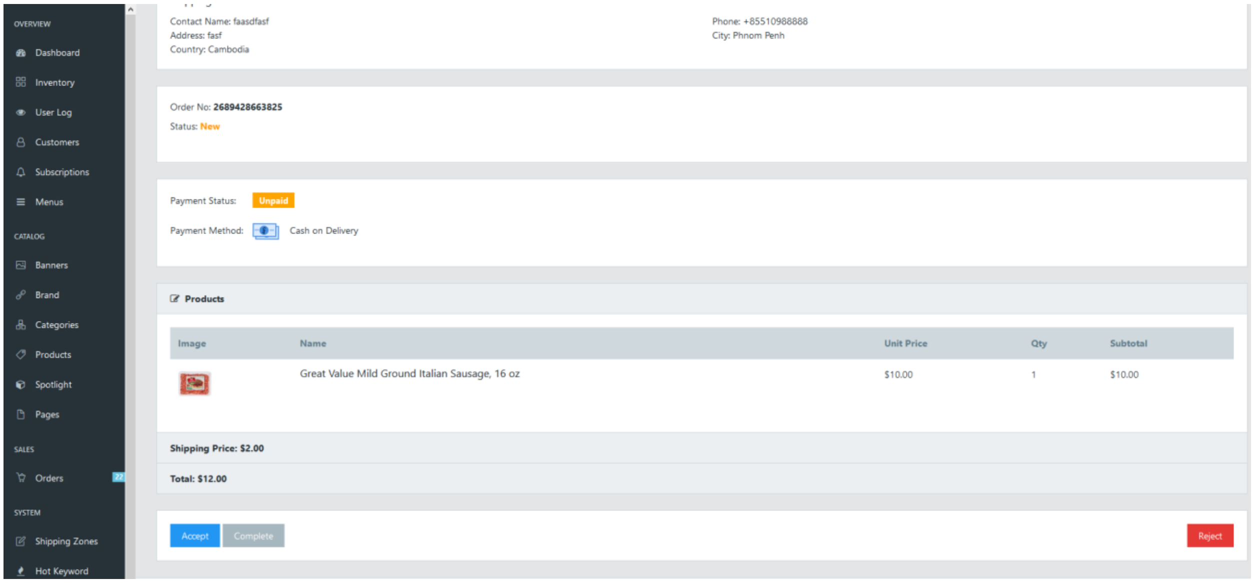


Order

16/06/2019 - 15/07/2019

Order No	Customer	Status	Payment Status	Payment Method	Total	Purchase Date	Complete Date	Action
2689428663825	blabla carla	New	Unpaid		$12.00	15/07/2019	-	✖
1827638816898	Admin	New	Unpaid		$1,069.00	28/06/2019	-	✖
2468258117487	Admin	Accepted	Unpaid		$735.00	27/06/2019	-	✖
2399533055913	Sor Seanghong	Rejected	Unpaid		$55.00	20/06/2019	-	✖

Appendix 9: Order List



Contact Name: faasdfasf
Address: fasf
Country: Cambodia

Phone: +85510988888
City: Phnom Penh

Order No: **2689428663825**
Status: New

Payment Status: Unpaid

Payment Method: Cash on Delivery

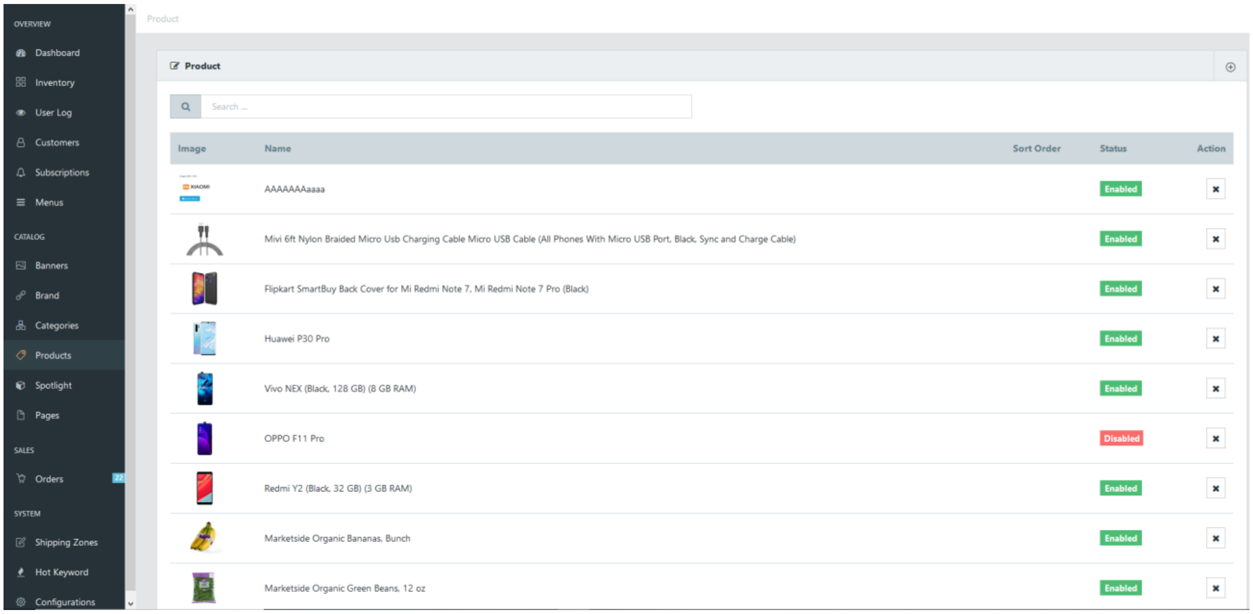
Products

Image	Name	Unit Price	Qty	Subtotal
	Great Value Mild Ground Italian Sausage, 16 oz	$10.00	1	$10.00

Shipping Price: $2.00

Total: $12.00

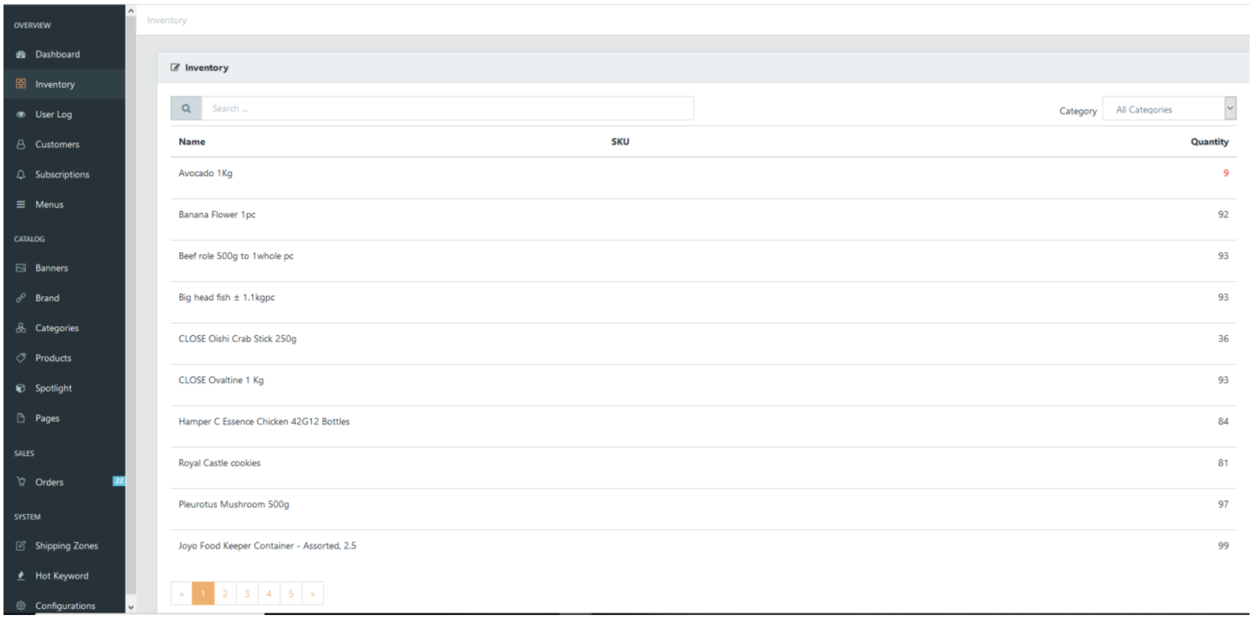
Appendix 10: Manage Order



Product

Image	Name	Sort Order	Status	Action
	AAAAAAAaaaa		Enabled	✖
	Mivi 6ft Nylon Braided Micro Usb Charging Cable Micro USB Cable (All Phones With Micro USB Port, Black, Sync and Charge Cable)		Enabled	✖
	Flipkart SmartBuy Back Cover for Mi Redmi Note 7, Mi Redmi Note 7 Pro (Black)		Enabled	✖
	Huawei P30 Pro		Enabled	✖
	Vivo NEX (Black, 128 GB) (8 GB RAM)		Enabled	✖
	OPPO F11 Pro		Disabled	✖
	Redmi Y2 (Black, 32 GB) (3 GB RAM)		Enabled	✖
	Marketside Organic Bananas, Bunch		Enabled	✖
	Marketside Organic Green Beans, 12 oz		Enabled	✖

Appendix 11: Product List

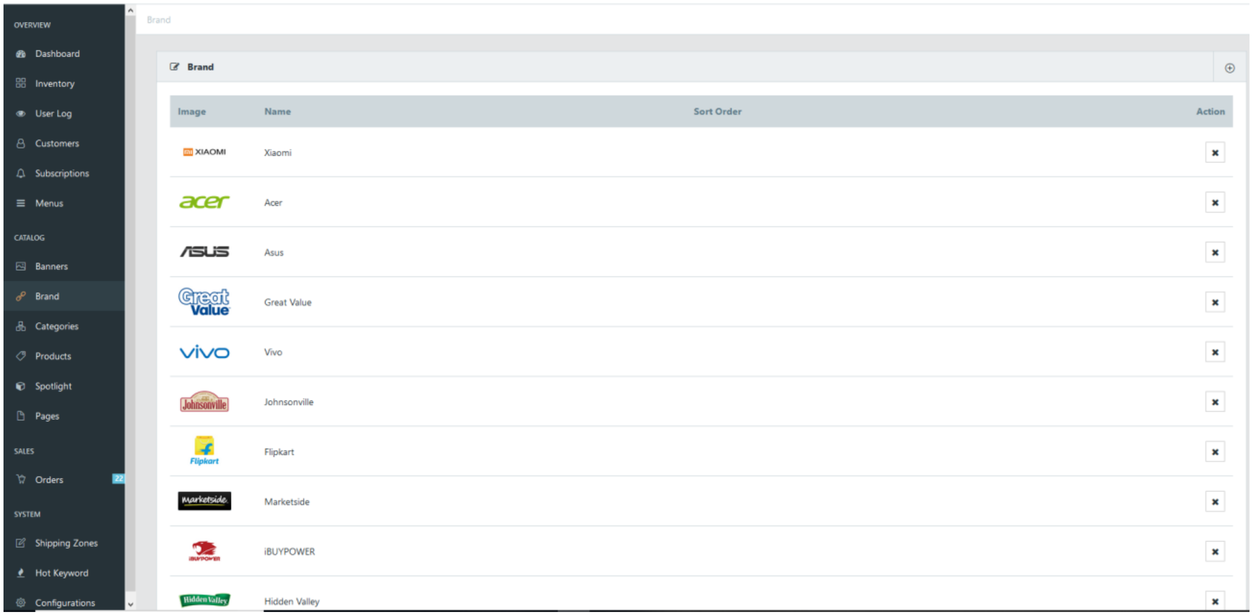


Inventory

Category: All Categories

Name	SKU	Quantity
Avocado 1Kg		9
Banana Flower 1pc		92
Beef role 500g to 1whole pc		93
Big head fish ± 1.1kgpc		93
CLOSE Oishi Crab Stick 250g		36
CLOSE Ovaltine 1 Kg		93
Hamper C Essence Chicken 42G12 Bottles		84
Royal Castle cookies		81
Pleurotus Mushroom 500g		97
Joyo Food Keeper Container - Assorted, 2.5		99

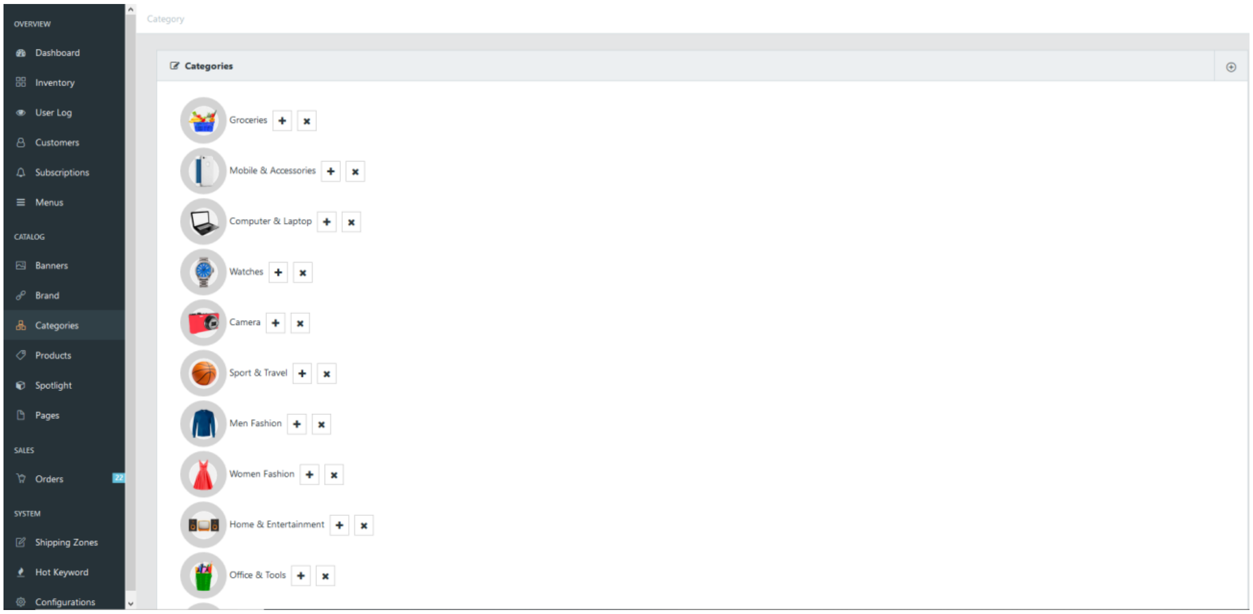
Appendix 12: Stock Management



Brand

Image	Name	Sort Order	Action
XIAOMI	Xiaomi		✖
acer	Acer		✖
ASUS	Asus		✖
Great Value	Great Value		✖
vivo	Vivo		✖
Johnsonville	Johnsonville		✖
Flipkart	Flipkart		✖
Marketside	Marketside		✖
iBUYPOWER	iBUYPOWER		✖
Hidden Valley	Hidden Valley		✖

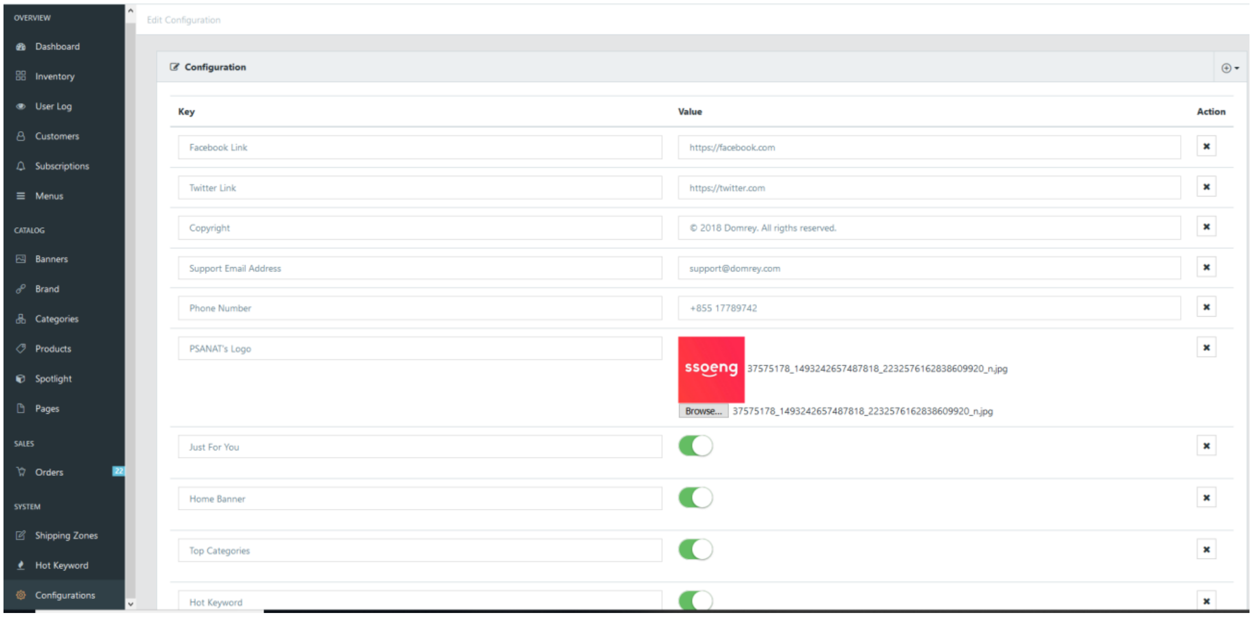
Appendix 13: Brand Management



Categories

- Groceries ✚ ✖
- Mobile & Accessories ✚ ✖
- Computer & Laptop ✚ ✖
- Watches ✚ ✖
- Camera ✚ ✖
- Sport & Travel ✚ ✖
- Men Fashion ✚ ✖
- Women Fashion ✚ ✖
- Home & Entertainment ✚ ✖
- Office & Tools ✚ ✖

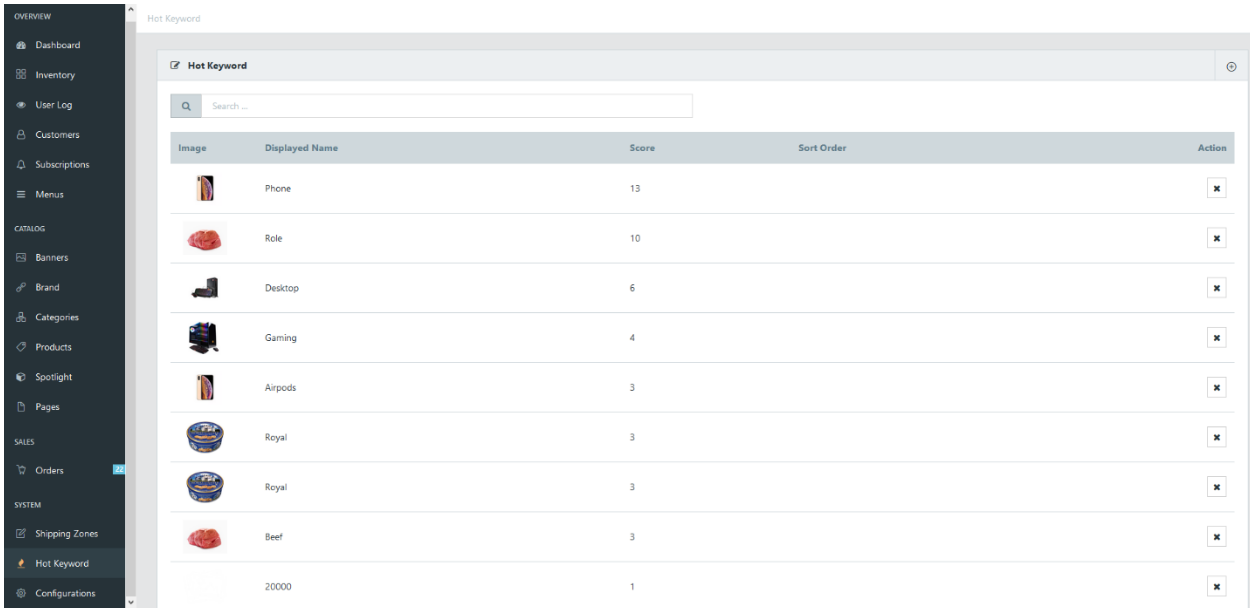
Appendix 14: Category Management



Configuration

Key	Value	Action
Facebook Link	https://facebook.com	✖
Twitter Link	https://twitter.com	✖
Copyright	© 2018 Domrey. All rigths reserved.	✖
Support Email Address	support@domrey.com	✖
Phone Number	+855 17789742	✖
PSANAT's Logo	ssoeng 37575178_1493242657487818_2232576162838609920_n.jpg 37575178_1493242657487818_2232576162838609920_n.jpg	✖
Just For You	●	✖
Home Banner	●	✖
Top Categories	●	✖
Hot Keyword	●	✖

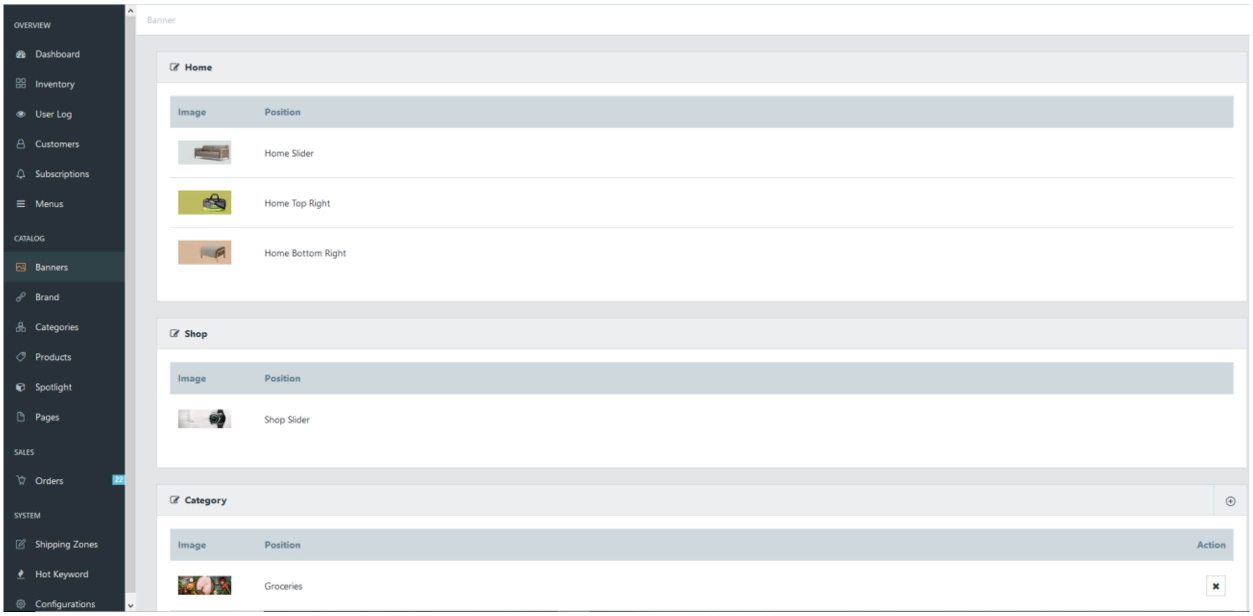
Appendix 15: Configuration Management



Hot Keyword

Image	Displayed Name	Score	Sort Order	Action
	Phone	13		✖
	Role	10		✖
	Desktop	6		✖
	Gaming	4		✖
	Airpods	3		✖
	Royal	3		✖
	Royal	3		✖
	Beef	3		✖
	20000	1		✖

Appendix 16: Hot Keyword Management



Home

Image	Position
	Home Slider
	Home Top Right
	Home Bottom Right

Shop

Image	Position
	Shop Slider

Category

Image	Position	Action
	Groceries	✖

Appendix 17: Banner Management