

# Design and Implementation of an Online Shopping Web Application

# ——Dubai Shop

Student: Jiang Chufeng

Student No.: 2378164

Supervised by Ahmad Ibrahim

MSc Computer Science

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham Dubai

September 2022

## Abstract

This is a full stack development project and the goal of the project is to create and develop a real-world website that mimicked Amazon.com,Inc. and to develop a comprehensive solution that provided customers with a variety of purchasing choices. It is an e-commerce website designed with Java, JS, HTML, CSS and Spring Boot, named "Dubai Shop", where customers can shop online.

"Dubai Shop" is a complete solution for customers seeking to purchase goods, and provides an all-in-one content management system for backend users such as administrators, salespeople, and shippers who are responsible for managing the frontend content and issues of the system.

This web application is based on MVC architecture, integrated with Spring Data JPA with Hibernate to handle data access, Spring Restful Web Services returning JSON responses in conjunction with jQuery, and Thymeleaf for creating dynamic frontend views, and so on.

The project has resulted in the development of a working prototype of an online shopping website at the completion of the project.

Key words: HTML; CSS; JS; Bootstrap; MySQL; Spring Boot

## Acknowledgements

This project has been carried out under the supervision of Dr. Ahmad Ibrahim and under the inspection of Dr. Eike Ritter, and I would like to thank them for providing guidance and feedback throughout the project.

In addition, I would like to take this opportunity to thank my parents, Jiang Yuying and Jiang Zhaosheng, who have been unconditionally supporting me throughout this very challenging academic year.

Furthermore, it is also my privilege to have Dr. Mian M. Hamayun, Dr. Ahmad Ibrahim, Dr. Anis Zarrad, and Dr. Kashif Rajpoot provided me with an in-depth and comprehensive understanding of computer science through their courses in full stack development, computer system, data structures and algorithms, databases, systems design, and artificial intelligence during this academic year.

Finally, I would like to thank my classmates in Dubai for their kindness and help through both academic and personal difficulties. There is nothing better than studying with you all during the year, we did our study together, we went to the cheapest PCR Clinic together, we collaborated on debugging together, you alerted me when a credit card fraud occurred, and you took me home whenever you can. Studied with you all did make it quite a special year

## Introduction

In today's online shopping environment, multichannel selling is a significant revenue possibility that cannot be ignored, especially since it is growing at a rapid pace. As the number of sales channels skyrockets, the process of fulfilling ecommerce orders becomes increasingly complex.

In today’s marketplace, there are multiple vendors, providers, or systems used by online shopping brands to manage fulfilments. In addition to posing a significant threat to productivity, this complexity of fulfillment workflows, guidelines, and service providers can also result in unnecessary costs and insufficient operational transparency. It is possible to miss out on a significant number of potential customers by relying solely on brick-and-mortar stores. An online store is a great way for a business to grow and acquire new customers. A physical store will eventually reach a point where they will no longer be able to grow without a streamlined ecommerce model to maximize its potential.‍

In this report, I will describe how I developed the web application to manage the content management system from the backend and to manage the online shopping from the frontend. In hopes of providing a one-stop shopping solution for end users of the web application, it aims to provide a one-stop online shopping experience for customers.

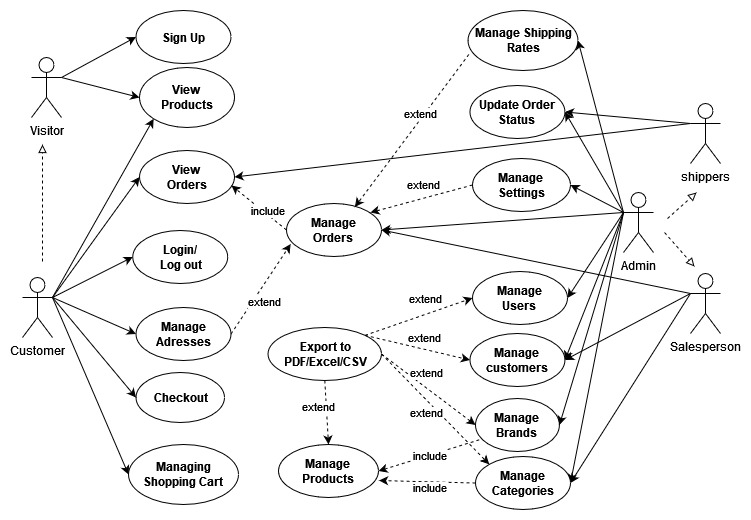
“Dubai Shop” as a parent includes two parts which are backend and frontend, and for the two parts

Report overview

## Project Specification

### 2.1 User Requirements

There are five different types of roles divided into frontend and backend users based on their functionality. The roles on the backend are the administrator who manage everything related to the website, the salesperson who handle the sales of products, and the shipper who deliver orders. Visitors and customers are the two different types of roles that reside in the frontend section.



### System Requirements

#### 2.2.1 Functional Requirements

1. The backend management system MUST retain information of every backend user including their:

|  |  |  |
| --- | --- | --- |
| * First and last name | * roles | * Email |
| * photo | * passwords |  |

1. The backend management system MUST retain information of products:

|  |  |  |
| --- | --- | --- |
| * Brands | * Brands ID | * Brand Logo |
| * Product Categories | * Product Images | * Prices |

1. The backend and frontend systems MUST retain information of orders:

|  |  |  |
| --- | --- | --- |
| * Order ID | * Product ID | * Quantities |
| * Customer ID | * Customer Address | * Customer Name |
| * Prices | * Shipping Cost | * Order Status |
| * Order Time | * Deliver Date | * Payment Method |

1. The frontend system MUST retain information of customers:

|  |  |  |
| --- | --- | --- |
| * Customer ID | * Customer Address | * Customer Name |
| * Customer Phone | * Customer password | * Customer type |

1. The frontend system MUST retain information of shopping carts:

|  |  |  |
| --- | --- | --- |
| * Customer ID | * Customer Address | * Customer Name |
| * Product ID * Product Price | * Shipping Cost | * Quantities |

1. The backend and frontend systems users CAN edit their personal information.
2. The backend and frontend systems MUST give out alerts when backend users and frontend customers input illegal format data or click wrong button.
3. The backend system MUST distinguish different types of users and give different authorities to access different modules.
4. The frontend system MUST display the total value of the selected items with a single line of tax charged, and shipping costs separately.
5. The frontend users MUST be able to view the categories on the home page.
6. The frontend users MUST be able to view items in different categories or different brands.
7. The customer SHALL be able to view information about an item before adding it to the cart.
8. The visitor MUST be able to register as a customer.
9. The customer MUST be able add goods to the cart.
10. The customer MUST be able to able view the shopping cart.
11. The customer MUST be able to check out goods selected in the cart.
12. The visitor MUST register using the E-mail verification code.
13. The visitor MUST register before adding goods to the shopping cart.
14. The customer CAN NOT login or register if the information is incomplete or invalid
15. The customer CAN NOT place an order without completing the order form.
16. The administrator MUST be able to view all the frontend and backend users’ information.
17. The administrator and the salesperson MUST be able to add new goods to customers’ shopping items.
18. The administrator and the salesperson MUST be able to edit a product’s price, images and description.
19. The administrator and the salesperson MUST be able to delete goods from the customer’s orders.
20. The salespersons SHALL be able to view all the customers’ order details after they complete the order form and finish the checkout process.
21. The salespersons MUST be able to edit an product’s price, images and description.
22. The shipper MUST be able to view customer’s orders.
23. The shipper MUST be able to search orders placed.
24. The shipper MUST be able to update customer orders’ current status (picked, shipping or delivered)

#### 2.2.2 Non-Functional Requirements

1. Frontend and backend users MUST be able to access the system all year round any time, 24/7.
2. The website MUST be accessible from any devices connected to the Internet for example PC, tablet, and mobile phone.
3. Passwords MUST be encrypted rather than stored in plain text in the databases.
4. The system SHALL be able to save goods in the shopping cart if the website close unexpectedly.
5. The system MUST keep the frontend and backend users’ data persistent when the system get updated.
6. The website MUST be easy and simple to use and does not require much training for all customers.
7. The customer SHALL be able to add a product to the shopping cart in fewer than 4 seconds.
8. The customer SHALL be able to view information about a product in fewer than 4 seconds.
9. The customer SHALL be able to check out the goods in the shopping cart within 20 seconds after making payment.
10. The shift between pages SHALL take fewer than 2 seconds.
11. The system SHOULD have scalability for developing new features on demand and implementing well under high load.
12. The backend and frontend applications SHALL be scaled separately.

### 2.3 Use Cases Scenarios

#### 2.3.1 Visitor Purchasing

[Precondition]

1. A visitor is browsing the website on a laptop and wants to purchase a camera.
2. The visitor has his own phone number and E-mail.

[Flow of Events]

1. The use case starts when the visitor accesses the “Dubai Shop” website's home page.
2. The visitor clicks “Register” button to register as a customer:
   1. The visitor turns to the register page and fill in the register form.
   2. The visitor clicks “Create Account” button.
   3. The system sends a verification E-mail to the visitor’s account.
   4. The visitor opens his E-mail and receives the verification letter.
   5. The visitor clicks the verify link in the verification letter,
   6. The visitor’s account is enabled, and he successfully registered as a customer.
3. The customer searches for the desired product by clicking the icon in the home page: he clicks “Digital Cameras”, and then get into products display page.
4. After looking through descriptions for different types of cameras, the customer finds the suitable product and takes the desirable camera into his shopping cart.
5. If the customer leaves the cart, the camera stays in the cart. Next time when the customer come back, he can go to checkout or go for shopping the second time.
6. After the camera is added into the cart, the checkout process begins:
   1. The customer is required to check the shipping address whether it is support for shopping. If the Address is not support for shipping, the customer is required to modify his address or add another address in his address book.
   2. The total purchase amount is displayed including the taxes and the shipping fees. The customer is then presented with Cash/Paypal/Credit Card payment method to choose from.
   3. The customer selects the payment method provided and fill in the required information.
   4. If the payment successfully issued, then the payments is completed, otherwise he will return to the shopping cart.
7. A message displays that the order is successful launched.
8. The system sends an order confirmation to the customer’s E-mail address.
9. The order status automatically updated as “PAID”
10. The details regarding the order are sent to the Salesperson.

[Postcondition]

1. The order has been placed.
2. The customer is waiting the salesperson to deliver and the shipper to dispatch the camera.

#### 2.3.2 The salesperson and the shipper Processing

[Precondition]

1. The salesperson received the order to buy camera.
2. The shipper is available
3. The product is available.

[Flow of Events]

1. The salesperson receives the order.
2. The salesperson picks up the camera in his warehouse and packs the camera.
3. The salesperson contacts the shipper to take the package.
4. The salesperson delivers the package to the shipper.
5. The shipper updates the order’s status as “PICKED”.
6. The shipper updates the order’s status as “SHIPPING”.
7. The shipper takes the package to the customer’s address and gives it to the customer.
8. The shipper updates the order’s status as “DELIVERD”.
9. The salesperson updates the orders status to “Shipped”.

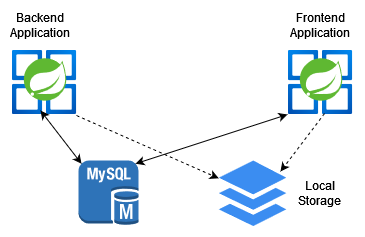
[Postcondition]

1. The customer received the camera.
2. Salesperson has successfully sold out a camera if the customer is not requiring of a return.
3. The shipper has completed an order.

## Solution Design

### 3.1 System Architecture

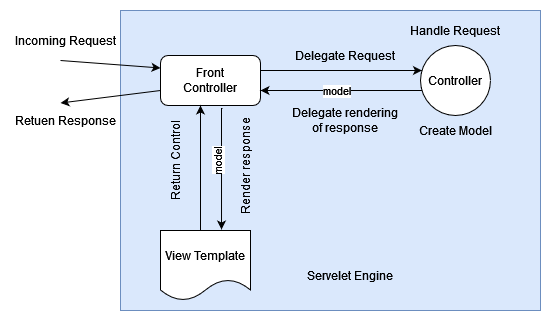
“Dubai Shop” is running on a local development environment in a single computer. This Web Application includes Backend application and the Frontend application, both running as a standalone JAR files. Spring boot Jars are self-contained Jar that has a collection of all the dependencies required for running the application (EDUCBA, 2022). The backend application and frontend application are connecting with local databases MySQL, and the static resources such as site logo, product images, brand images and category images are stored in the local environment.



Although the Backend application and the Frontend application are separate applications, they have some shared code for common entities, as they are both supported by the same database. As well as their own dependencies, both Spring Boot web applications share several common dependencies, such as Spring Data JPA, MySQL JDBC driver, and Spring Web. Therefore, there are three coding components to this project: Backend code, Frontend code, and Common codes shared between both applications.

### High-level Architecture (MVC)

The MVC architecture will be used to develop this project because I can practice the skills I learned in my *Full-Stack Application Development* Module in my second semester. Model–View–Controller is a popular software pattern used to break up the logic of your application into three different components ([Jessica Wilkins, 2021)](https://www.freecodecamp.org/news/author/jessica-wilkins/) . A spring MVC is a certain kind of container of the Spring framework and Model View Controller, which is easy to use to build web application, implementing core spring framework features like IOC and dependency injection (Scaler Academy, 2022).



Source: Spring Framework Reference Documentation, Rod Johnson, et al 2016

According to the “Spring MVC Tutorial” posted on the javatpoint website, introduction for the concepts of MVC terms are as follow:

* ***Model*** *- A model contains the data of the application. Objects can be single or grouped together to make up a data object.*
* ***Controller*** *- Business logic is contained within a controller.A class is marked as a controller by adding the @Controller annotation.*
* ***View*** *- In a view, information is presented in a specific format. View pages are generally created using JSP with JSTL. Additionally, spring supports Apache Velocity, Thymeleaf, and FreeMarker view technologies.*
* ***Front Controller*** *– The Servlet Dispatcher is the front controller whose job is to handle dispatches of requests in Spring Web MVC. This component manages the flow of Spring MVC applications.*

Regarding "Dubai Shop", this project utilizes the MVC architecture as well. The view layer renders the HTML web pages to the client using thymeleaf and HTML code, such as order.html, order\_form.html, and address\_book.html. The view layer invokes the controller layer that uses the Spring MVC Controller, or REST controller or RESTful webservices to hand the request from the clients. The controller layer calls the service layer to perform the business logic of the application. And then it will render the view that is returned to the client. The service layer depends on the repository layer that contains entity classes and repository interfaces. Below the repository is the Spring Data JPA layer which uses Hibernate framework as the implementation of Spring Data JPA. Furthermore, the Hibernate uses JDBC driver to communicate with the underlying database instance.

图示

描述已自动生成

### Technology Stack

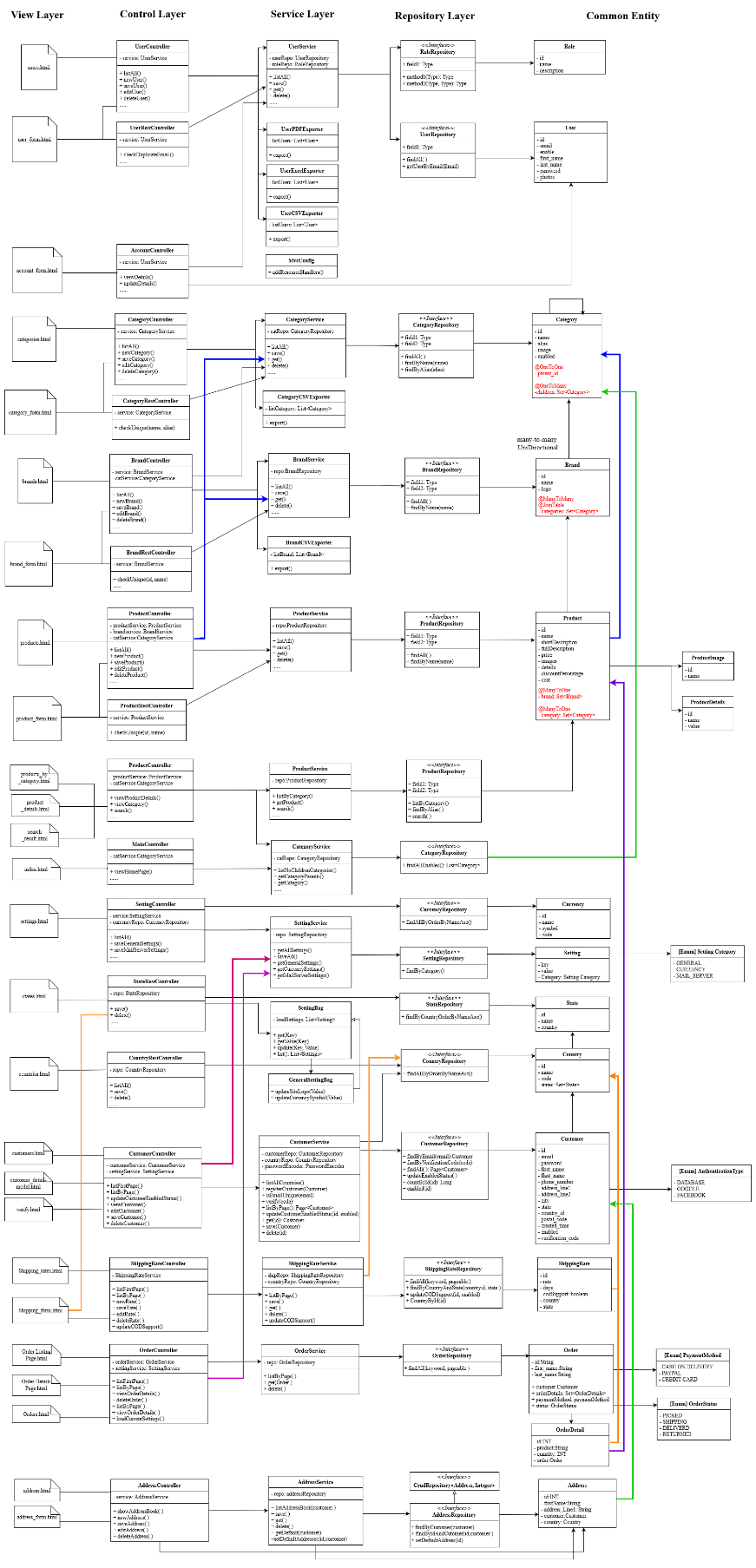
### Entity Relationship Diagram

图示

描述已自动生成

### Class Diagram

#### Admin Management Side



#### Customer Shopping Side

图示, 日程表

描述已自动生成

### Activity Diagram

#### 3.6.1 Visitor Registration and Customer Login

图示

描述已自动生成

#### 3.6.2 Customer Purchasing and Shipper Delivering

图示

描述已自动生成

### Sequence Diagram

#### Visitor Registration

图示

描述已自动生成

#### Purchasing products

图示

描述已自动生成

#### Shipping Products

图示

低可信度描述已自动生成

### Activity Diagram

## Solution Implementation

### 4.1 Admin User Module

#### 4.1.1 Encoder User Password

In the “Dubai Shop” Project, the BCrypt password encoder was used to help secure the password security. It is provided by Spring Security and is a password-hashing function based on Blowfish and crypt function in Unix (Wikipedia, 2022). The hashed password stored in the database is the result of encrypting the value "OrpheanBeholderScryDoubt" 64 times with the final state from the last run of the key schedule (Rishi Raman, 2021)

Following is a description adopted from an article written by Emmanuel Hayford (2021) of how hash-functions enable authentication for admin management system:

1. A user named Jiang Chufeng, creates an admin control panel account.
2. Later, Jiang Chufeng chooses a username and password for herself.
3. The server performs a Bycrypt algorithm operation on Jiang Chufeng’s password, keeps the hashed password, and stores the string in the database.
4. when Jiang Chufeng returns to the site, her login details are securely transferred to the system’s backend.
5. Once the backend server receives Jiang Chufeng's login information, it searches the database and sends her password through the same hash function she used when she registered.
6. If the password hash of the password Alice entered matches the one stored in the database, Alice is granted access. Otherwise, the site denies Alice access.
7. Access is granted to Jiang Chufeng if the password hash matches the one stored in the database. Otherwise, she will not be able to access the system.

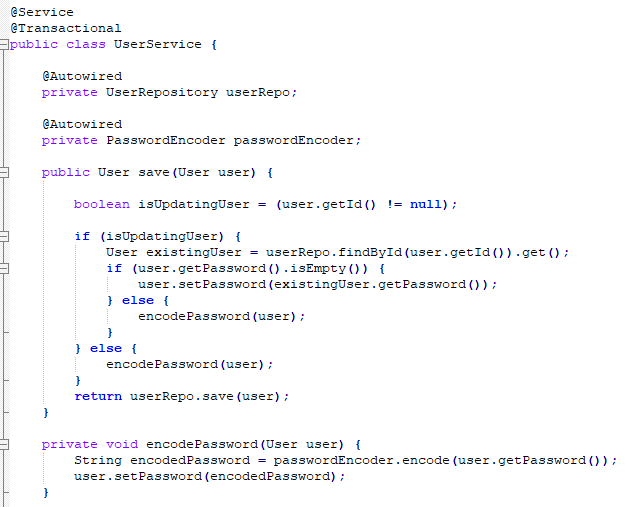
Let me explain how the code is play in this project.

Using the Bcrypt algorithm, BCryptPasswordEncoder implements the hashing of passwords. Define the BCryptPasswordEncoder as a bean in the project backend module’s WebSecurityConfig java class file:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

To encode the password when registration or edition, the PasswordEncoderin in the UserService hashed the password input. Here is how the code is doing:



Above is the code to encode the password before the database storing it. Here are the main steps to achieve this:

1. We inject password provider in the method of authentication provider.

2. We inject the UserService class.

3. The custom authentication provider is used to encode the password.

Apart from the user module in the Admin Management System, frontend customers’ passwords are also encrypted and stored in the databases following the same way using BCryptPasswordEncoder.

#### 4.1.2 User Authentication

This project has the User entity class, the UserRepository interface, and WebSecurityConfig class that configures custom login page. Apart from that, a DubaiShopUserDetails class implements the UserDetails interface that defined by Spring Security, and this class wraps an instance of User object. The DubaiShopUserDetailsService, which implements the UserDetailsService defined by the Spring Security, is created to perform user login authentication. Once successful authentication, a DubaiShopUserDetails instance representing the currently logged in user will be returned by Spring Security. In the DubaiShopUserDetailsService class, the *loadUserByUsername()* method *is* overrided and calls the UserRepository's *getUserByEmail()* method that returns a User based on the email. The Spring security will validate the user's password based on the returned object to perform authentication. The WebSecurityConfig class is to inform Spring Security to use DubaiShopUserDetailsService as a UserDetailsService, and the authentication provider is database-based. At the same time, some filters in the Spring Security Filter chain behind the scenes will intercept the request from the frontend page, and the request will call the UserDetailsService to perform authentication.

图示

描述已自动生成

In the Backend, different types of users have different access to certain pages. Here is the code.



In the Frontend, security expression “hasAnyAuthority(…)” is used to give specific access right to different users.

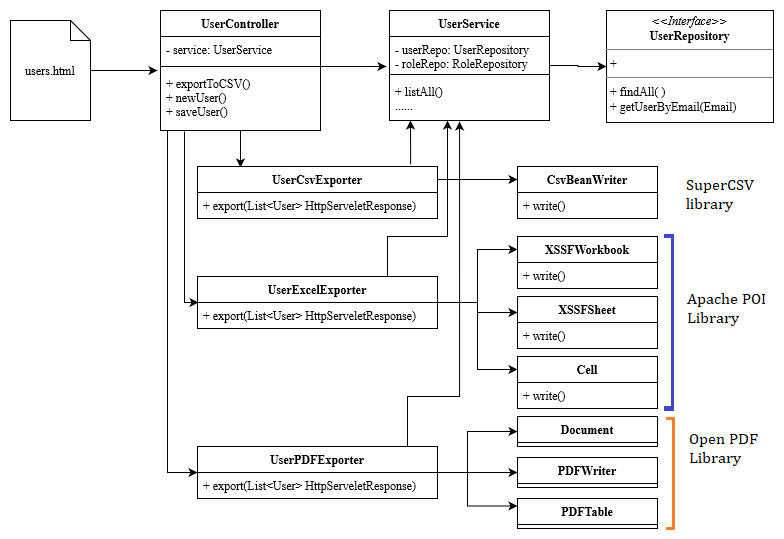
文本, 信件

描述已自动生成

#### 4.1.3 Export to CSV/EXCEL/PDF

Handler methods such as *exportToCSV(), exportToExcel()* and *exportToPdf()* in the UserController is to handle the hyperlink in the frontend to export different types of files. The business classes such as UserCsvExporter, UserExcelExporter, and UserPdfExporter read the user information from a List collection of User objects and write data to a HttpServletResponse. Therefore, the fronted user can download data as CSV/EXCEL/PDF from the browser.

For CSV file, CsvBeanWriter is used, and this class is from the SuperCSV library - a free and open-source Java library to read and write CSV files (Apache, 2015). For Excel documents, some API from the Apache POI Library are used. Apache POI is a free and open-source Java library to read and write Microsoft Office documents (The Apache Software Foundation, 2022). For PDF documents, some APIs from the Open PDF Library are used, and it is also a free and open-source Java Library to process PDF documents (Abdullah Çevik, et al. 2022).

****

### Setting Module

#### 4.2.1 Setting Filter

In the setting module, Java filter is the most important part to apply for the entire shopping application. Using the setting filter, all requests coming into the application are intercepted by the application before being handled by any controller. A view will be dynamically updated with the setting values after the system reading the setting values from the database and putting them on the model.

图示

描述已自动生成

we use the service class to read the settings information from the database and introduce you doFilter() method. We put the setting values into a request and continues the filter chain. we have the same attitude that intercepts also requests coming to the application and the filter we modify the request by putting setting value into as our request before the requests are handled by the controllers and in the view layer. We apply the setting values for the view files before returning the response to the client by using filter.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

#### 4.2.2 Load Country with AJAX and Restful Webservice

ALL AJAX Calls to the RESTful Webservices are performed in loading and managing Country settings.

The unique features of AJAX make it the dream of every developer, as it can be used to read data from a web server after the web page has loaded, to update a web page without reloading, and even to send data to the web server in the background (W3Schools, 2022).

图示

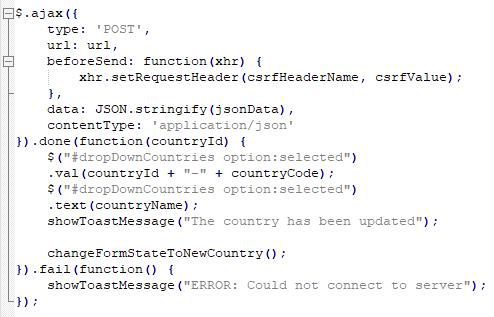
描述已自动生成

The REST architectural style is an approach to designing Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP using a variety of clients. The REST protocol requires developers to use HTTP methods explicitly and in a consistent manner. The basic REST design principle establishes a one-to-one mapping using *POST* annotation to create a resource on the server, using *GET* annotation to retrieve a resource, using *PUT* annotation to update the resource, and using *DELETE* annotation to remove a resource (Alex Rodriguez, 2015).

图示

描述已自动生成

In the “DubaiShop” project, some Restful Webservices are used by the client with some JavaScript and jQuery codes to send AJAX requests to perform the REST Webservices. And data will be sent along with the requests and responses when necessary. When the settings pass is loaded, it does not immediately load the countries since there are more than 200 countries in the database. The country list will be only loaded when the user clicking the “Load countries” button. Country data is used in customers module, address module, shipping rates module, and checkout module.



In Spring, the objects are automatically converted into JSON data which is sent to the client. For creating and updating operation, a *@PostMapping* annotation is performed. For HTTP posting requests, the *@requestBody* annotation is used to convert JSON data from the client into a Java object. And for delete operation, the *@DeleteMapping* annotation is used for HTTP delete method.

文本

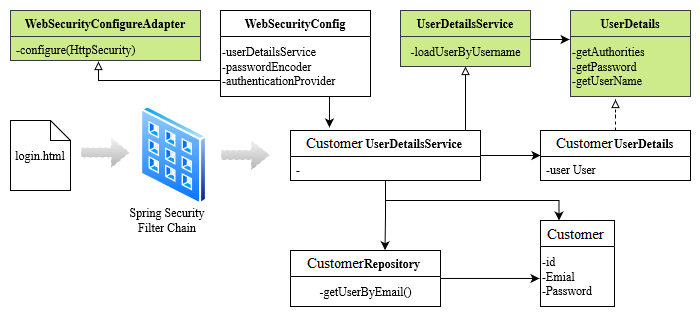
描述已自动生成

### 4.3 Cutomer Module

#### 4.3.1 Customer Authentication

Customer authentication in the customer module follows the same logic with the feature of user authentication in the Admin Management System.

A DubaiShopUserDetails class implements the UserDetails interface that defined by Spring Security, and this class wraps an instance of User object. The CustomerUserDetailsService, which implements the UserDetailsService defined by the Spring Security, is created to perform customer login authentication. Once successful authentication, a CustomerUserDetails instance representing the currently logged in customer will be returned by Spring Security. In the CustomerUserDetailsService class, the *loadUserByUsername()* method is overrided and calls the UserRepository's *getUserByEmail()* method that returns a Customer based on the email. The Spring security will validate the customer's password based on the returned object to perform authentication. The WebSecurityConfig class is to inform Spring Security to use CustomerUserDetailsService as a UserDetailsService, and the authentication provider is database-based. At the same time, some filters in the Spring Security Filter chain behind the scenes will intercept the request from the frontend page, and the request will call the UserDetailsService to perform authentication.



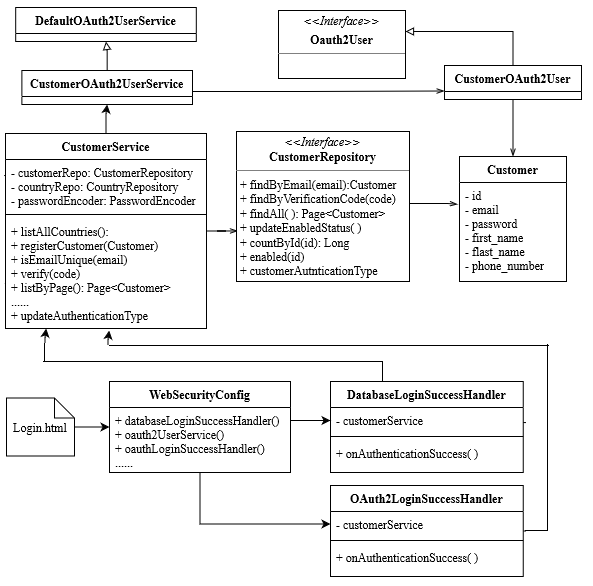
Codes of customer authentication is adapted from Admin Management System’s user authentication module. Therefore, no further elaboration here.

#### 4.3.2 Google and Facebook Login

In the “DubaiShop” project, OAuth2.0 standing for “Open Authorization” is used to help visitors register or login as a customer. Authorization is carried out using the OAuth 2.0 protocol, which is considered the industry standard. Web, desktop, mobile, and living room applications can all benefit from OAuth 2.0, which emphasizes client developer simplicity ([IETF OAuth Working Group](https://www.ietf.org/mailman/listinfo/oauth), 2013).

Client credentials for Google OAuth2 authentication can be obtained from the [Google API Console](https://console.developers.google.com/). In the “Credentials” section, “OAuth2 Client ID” for “DubaiShop” web application is created. As a result, Google sets up a client id and secret for the developer.

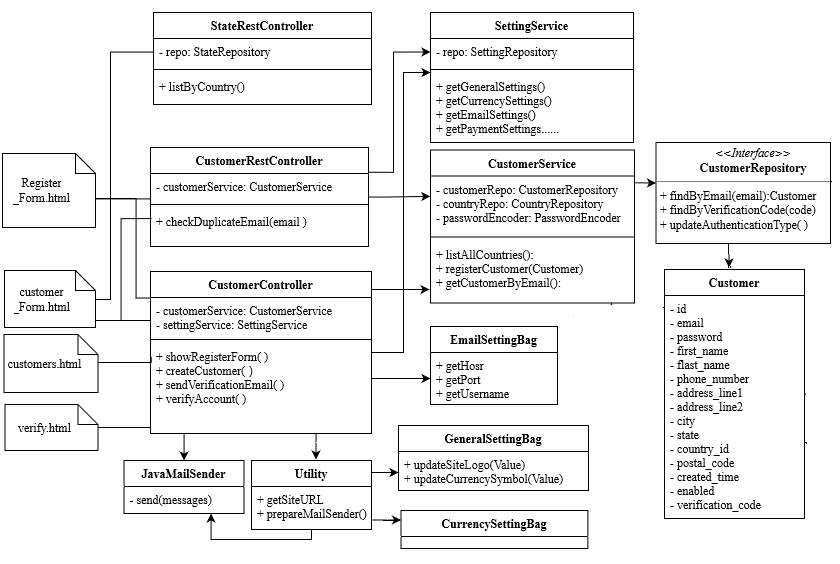
In the coding section, the customerOAuth2Users class inherited from OAuth2User class defined by Spring Security, and it is a client library to represent user authenticated through OAuth protocol. In addition, customerOAuth2Users class also called the customerOAuth2UserService class which inherited from the DefaultOAuth2UserService. This class uses a load user method which will be invoked by Spring Security OAuth client library upon the OAuth to authorization.



#### 4.3.3 Customer E-mail Verification

At the time submit the signup form, a visitor will receive a verification code, and the visitor should have to click on the verification hyperlink in order to activate his personal account and to become a customer. Then the visitor can login and making payment.

The User entity class has two fields, verificationCode and enabled. The verificationCode is a unique string generated randomly during the registration process and is used in the verification process. The CustomerUserDetails class overrides the *isEnabled()* method which returns the user’s enabled status. Therefore, if the visitor has not verified his account, the Spring Security will inject login. The UserServices class uses the JavaMailSender and have *register()* and *sendVerificationEmail()* methods to help send an verification email to the visitor’s inbox, with the verification hyperlink. In case a user has not activated his registered account by clicking on the verification link embedded in his email, his enabled status will be false, and the user will not be able to log in ([Nam Ha Minh](https://www.codejava.net/nam-ha-minh), 2020).



### 4.4 Checkout Module

#### 4.4.1 Checkout and Send Order Confirmation E-mail

The CheckoutService implements the business methods in Checkout Controller such as *prepareCheckout()*, *calculateProductCost( )*, and *calculateShippingCost( )*. To validate the order, the PapalService class is used for help, and Paypal function will be discussed in the later paragraph. The Chckout Controller implements the handler methods such as *processPaypalOrder()*, *showCheckoutPage()*, and *placeOrder().*This Checkout Controller uses several services classes, such as CustomerService class, OrderService class, AddressService class, SettingService class, ShoppingCartService, ShippingRateService, and CheckoutService class. The CheckoutInfo presents the checkout information in the view layer, such as Checkout page and ShoppingCart Page.

图示

描述已自动生成

When coding order E-mail confirmation, EmailSettingBag class is used to get the confirmation subject and the content stored in the databases. The SettingService class gets currency settings which details the using currency set by the Admin management System, and help determine the order currency unit in the confirmation E-mail.

图示

描述已自动生成

#### 4.4.2 Paypal Service

Get the client id and client secret of “DubaiShop” application by creating a PayPal developer account and creating an app in PayPal's sandbox. The PaypalOrder Response class is to map some values in the JSON response to a Java object with fields ID and status, and it also remains *validateOrder()* method in the Json response. The PaypalService class includes the *validateOrder()* method which will call the Checkout Controller class for validating a Paypal order. In the PaypalService class, the *@component* annotation is used instead of the *@service* annotation, because this class reference to the SettingService class to get the payment settings. The Checkout Controller is responsible for handling Paypal orders, which are invoked after payment is processed by the checkout-based wrapper

图示

描述已自动生成

### 4.5 User Interface

#### 4.5.1 Shopping Home

When visitor open his browser and open “DubaiShop” Website, The first thing go into his eyes is the shopping home page. Logo of the website is at the left-up corner, and there are five columns in the navigation bar which are Shopping, Payment, Shipping & Delivery, Login, and Register. At the center of this page, different categories of products was displayed in rows. If the visitor clicks the category icon or the category name under the icon, he will get into product page.

图形用户界面, 应用程序

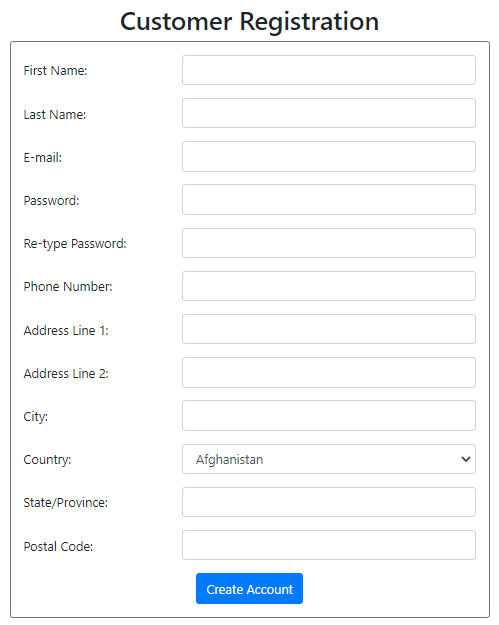
描述已自动生成

图形用户界面, 网站

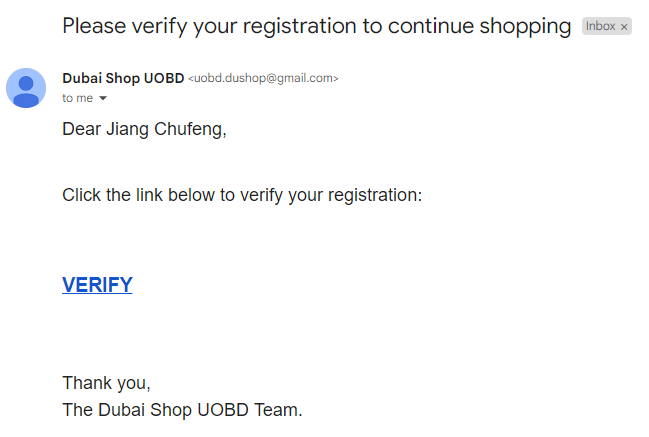
描述已自动生成

#### 4.5.2 Visitor Register and Customer Login

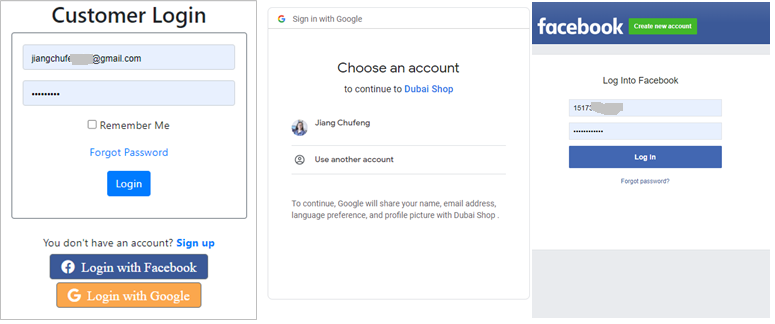
When the visitor clicks “Register”, he will get into the register page, and the register form will display in the screen. In this form, only the “Address Line 2” can be left empty, other blankets must be fill in completely, otherwise the “Create Account” button will not work.



Once the visitor fills out the form and click the “Create Account” button, a customer verification E-mail will be sent to his inbox. When the visitor clicks the “Verify” hyperlink in the confirmation letter, he will be redirected to the shopping website and a message will notify him that he has been registered successfully.

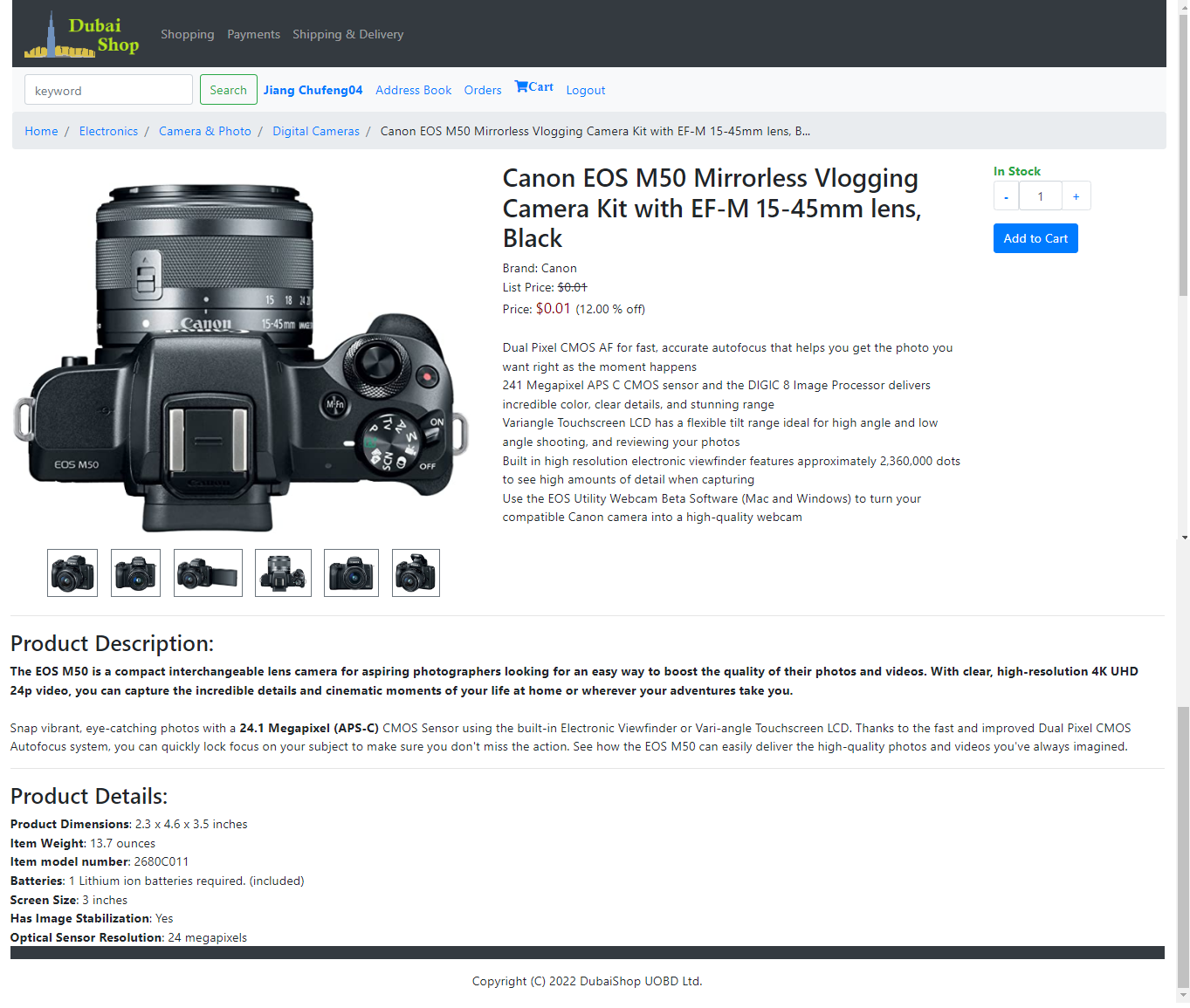


Once verified the account, the customer can login the website using the email address and password. Apart from register by email, the customer can login using his Google and Facebook account directly.



#### 4.5.3 Product Details

This is the Product display webpage. In this page, the product’s full name, description, specification, and images from different sides are included. If the customer clicks the main image of the product, the slide show will pop up. By clicking the arrows in the two sides, customer can look through the product into detail from different angles. If the customer is satisfied with the product, he can click the “Add to cart” button to put the selected product into shopping cart.



黑色的照相机

描述已自动生成

#### 4.5.4 Shopping Cart and Checkout

By clicking the “Cart” icon in the menu bar, the customer can jump to the shopping cart directly. In the shopping cart page, the customer can edit the quantity of products, and remove the unwanted products. Once decided the products to buy, the customer can click the “Chet Out” button, and the web page will redirect to the checkout counter.

图形用户界面, 文本, 应用程序

描述已自动生成

In the checkout counter, customers shipping information is displayed, and the order’s summary and costs are displayed at the right side of the screen. At the bottom of the page, customers can choose the payment method. If he chooses the “Cash on Delivery”, the order will be placed immediately. If he chooses “PayPal” or “Credit Card”, a new window will pop up requiring further payment information.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面

描述已自动生成

图形用户界面, 应用程序, Teams

描述已自动生成

Once the payment information is filled properly and the customer clicks the “Pay Now”, the payment will be processed through the server. If the payment was successful, an order confirmation will be sent to the customer’s inbox. Otherwise, a notification will pop up indicating the failure in payment.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本

描述已自动生成

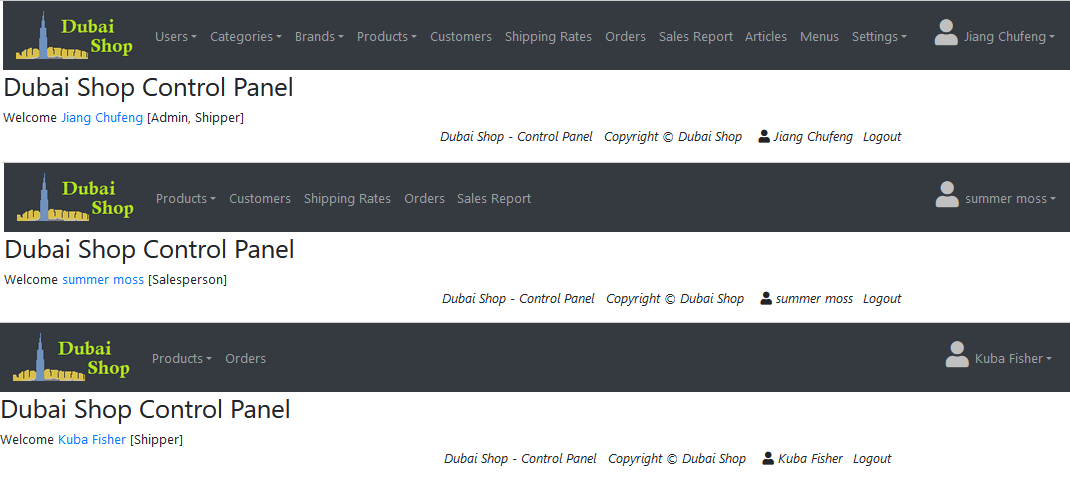
#### 4.5.1 Admin User Login

Users of the Admin Management System use another portal login the Control Panel. Different User have different interface after login.

图形用户界面, 应用程序

描述已自动生成

As shown in the FigureXXX, Admin user have all access to all modules, while shipper and salesperson have access to some of the modules, not full access. While if a limited authorization



4.5.1 Customer Login

#### 4.5.1 Admin User Login

#### 4.5.1 Admin User Login

#### 4.5.1 Admin User Login

## Reference

EDUCBA., (2022). “Introduction to Spring Boot Executable Jar.” [online] *EDUCBA.* Available at: <https://www.educba.com/spring-boot-executable-jar> (Accessed 03 September 2022).

[Jessica Wilkins, (2021).](https://www.freecodecamp.org/news/author/jessica-wilkins/)  “MVC Architecture – What is a Model View Controller Framework?” [online] *FreeCodeCamp.org*.

Available at: <https://www.freecodecamp.org/news/mvc-architecture-what-is-a-model-view-controller-framework/> (Accessed 06 September 2022).

Scaler Academy, (2022). “Difference Between Spring MVC and Spring Boot” [online] *InterviewBit.com* Available at:

<https://www.interviewbit.com/blog/difference-between-spring-mvc-and-spring-boot/> (Accessed 11 September 2022).

Rod Johnson, et al., (2016). “Spring Framework Reference Documentation”. [online] Available at:

<https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/index.html> (Accessed 11 September 2022).

Wikipedia, (2022). “Bcrypt”. [online] Available at:

<https://en.wikipedia.org/wiki/Bcrypt> (Accessed 13 September 2022).

Rishi Raman, (2021). “The Ultimate Guide to BCrypt and Authentication Protocols” [online] *clerk.dev* Available at: <https://clerk.dev/blog/bcrypt-hashing-authentication-encryption> (Accessed 13 September 2022).

Emmanuel Hayford (2021). “Understanding The bcrypt Hashing Function And Its Role in Rails.” [online] *manny.codes* Available at:

<https://manny.codes/understanding-the-bcrypt-hashing-function-and-its-role-in-rails/> (Accessed 13 September 2022).