



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Yang Ruan

THE DEVELOPMENT FOR AN ONLINE SHOP PLATFORM

School of Technology
2019

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Yang Ruan
Title	The Development For An Online Shop Platform
Year	2019
Language	English
Pages	43
Name of Supervisor	Prosi Pirjo

E-commerce is increasingly blooming and over the last decade it has increasingly been changing the lifestyle of people around the world. With this in mind the aim of this thesis was to analyze, design, implement and test a modern web application of a web shop based on Java and the SSM framework. The outcome of the thesis was a web application split into two sections, the user pages and the admin pages.

The user pages show a web shop selling electronic products. The user is able to login, register, and order products and then view their orders.

The admin pages allow admins to manage the application by being able to add, edit and delete information in the database.

The project was completed with all necessary requirements met.

Keywords Java, Web shop, Ecommerce

1 Contents

ABSTRACT	1
List of figures and tables	4
List of abbreviations	5
1. Introduction	6
2 Technologies used	7
2.1 HTML	7
2.1 CSS	7
2.2 JavaScript	8
2.3 JQuery	9
2.4 Java	9
2.5 J2EE	9
2.6 JavaBean	10
2.7 SSM framework	11
2.8 Eclipse	11
2.9 Mysql	12
3 Analysis	13
3.1 Requirements	13
3.1.1 Use Java technologies	14
3.1.2 Products page	14
3.1.3 Shopping cart functionality	14
3.1.4 Admin page	15
3.1.5 Login/register functionality	15
3.1.6 User friendly UI	15
3.1.7 Easy to understand error handling	15
3.1.8 Store user information securely	16
4 Design	17
4.1 Development process	17
4.2 UI Designs	18
4.2.1 Customer section	18

4.2.2	Admin section	20
4.3	Database design.....	21
5	Implementation	25
5.1	User interface implementation	25
5.2	Functionality	30
5.2.1	Outline	30
5.2.2	Functionality example	32
5.3	Database implementation	36
6	Testing	41
7	Conclusion	42
8	References.....	43

List of figures and tables

Figure 1. An example of CSS selector using classes	8
Figure 2. An example of a class using the JavaBean syntax	11
Figure 3. An example of MySQL usage showing a structure of a relational database.....	12
Figure 4. Mock design of UI for customer home page.....	18
Figure 5. ER diagram of the user table.....	21
Figure 6. ER diagram of the order table.....	22
Figure 7. ER diagram of item type table.....	22
Figure 8. ER diagram of board table	23
Figure 9. ER diagram of cart table.....	23
Figure 10. ER diagram of item table.....	24
Figure 11. Screenshot of the Home page.....	25
Figure 12. Screenshot of the login page.....	26
Figure 13. Screenshot of the cart page.	26
Figure 14. Screenshot of the orders page.	27
Figure 15. Screenshot of the profile page.....	27
Figure 16. Screenshot of the admin login page	28
Figure 17. Screenshot of the item management page	28
Figure 18. Screenshot of the item type page.	29
Figure 19. Screenshot of the notice management page.....	29
Figure 20. Screenshot of the order management page.....	29
Figure 21. Code snippet of the Javascript initTopList function.	33
Figure 22. Code snippet of ItemController showing topList method.	34
Figure 23. Code snippet of the ItemService interface.	34
Figure 24. Code snippet of the ItemServiceImpl class.....	35
Table 1. Table of requirements	13
Table 2. Implementation of the user table	36
Table 3. Implementation of cart table.	37
Table 4. Order table implementation.	38
Table 5. Item table implementation.	38
Table 6. Item type table implementation.	39
Table 7. Implementation of the board table	40

List of abbreviations

HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
SSM	Spring, spring MVC and Mybatis
MVC	Model-View-Controller
SQL	Structured query language
JDBC	Java Database Connectivity
POJO	Plain old java object
JSP	Java server pages
J2EE	Java enterprise edition
UI	User interface
JSON	Javascript object notation

1. Introduction

E-commerce is the trading of products electronically which is most cases done using the internet. It is increasingly blooming and over the decade has increasingly been changing the lifestyle of people around the world. In this sense, the project of a web shop based on SSM framework is an active response to the background of the current era. The outcome of this thesis will be a web application split into two sections, the user pages and the admin pages.

The user pages will show a web shop selling electronic products. The user will be able to login, register, order products and then view their orders.

The admin pages will allow admins to manage the application by being able to add, edit and delete information in the database.

2 Technologies used

2.1 HTML

HTML stand for Hypertext Markup Language, which is the now standard markup language used when developing web pages. The structure of the web pages is created using HTML tags for example, the paragraph tag (<p> </p>). The browser then renders the content of these said tags using the tag itself to decide how the content is displayed. The actual tags themselves are not displayed. Since 2015, HTML5 is the standard HTML version to be used. /1/

2.1 CSS

CSS stands for cascading style sheets and is used to define the style of a document written using a markup language for example HTML. The name comes from the fact that when there is conflicting styles defined, the last defined rule is the one to be used and so the priority ‘cascades’ down. There are three different ways in which CSS styles can be defined, an external style sheet, a style tag in the HTML page and inline style defined within a HTML tag. To target HTML elements for which to style, selectors are used. The three main selectors used are element tags, classes and ids. A class can be assigned to multiple HTML elements where as an id is only assigned to one element. Figure 1 shows an example of a CSS class defined style. /2/

Class Selector

```

<HTML>
<HEAD>
<style type="text/css">
simpleStyle {font-family:arial; font-size:14px;
color:blue}
</style>
</HEAD>
<BODY>
<b class="simpleStyle">Example of CSS Selector
usage. </b>
<br>
<br>
<i class="simpleClass">This is an italics tag carrying
the simpleStyle class</i>
</BODY>
</HTML>

```

Class Selector

Reusing same class selector for multiple HTML elements

Result:

Example of CSS Selector usage.

This is an italics tag carrying the simpleStyle class

Figure 1. An example of CSS selector using classes

2.2 JavaScript

JavaScript is a high level, interpreted programming language used along side HTML and CSS to develop web pages. JavaScript is prototype-based and dynamic, supporting different programming styles such as functional and object-oriented. JavaScript was at first only used in web browsers which each has its own JavaScript

engine for executing it. Now JavaScript is also used in web servers, databases and other software.

2.3 JQuery

JQuery is a free open-source JavaScript library which contains functions to make many common JavaScript tasks easier and simpler. JQuery focuses on DOM manipulation, event handling and Ajax.

2.4 Java

First releases in 1995 by Sun Microsystems Java is a programming language which enables developers to create applications on a wide range of platforms including desktops, datacenters, game consoles, mobile phone and the world wide web. Java uses a development environment called the Java development Kit, which is made up of an interpreter, a compiler and other tools which help create Java applications. The Java runtime environment, which is made up of many components including the Java virtual machine, is needed to run the Java applications. /3/

2.5 J2EE

Sun Microsystems along with other major companies including IBM and Oracle developed the J2EE middleware to create a set of guidelines to create and deploy scalable, multi-tier Java applications for businesses. As well as being scalable, other advantages of creating an application using J2EE are security, portability and reusability.

J2EE contains several layers:

1. Domain object layer. This layer consists of a series of POJO which stands for 'Plain Old Java Object', meaning it is just a pure data structure with no restrictions.
2. Data access object layer. This layer completes operation for database.
3. Business logic layer. This layer is the key to implement system functions.
4. Controller layer. This layer mainly calls methods provided by business logic component to handler user's request and then send result to presentation layer.
5. Presentation layer. This layer show the result from Controller layer. /4/

2.6 JavaBean

Wrote in the Java language, JavaBeans are classes used to encapsulate multiple objects into a singular object (a bean). Encapsulating multiple objects as one single object means it is easier to pass the objects around the program and create and modify in a build tool. The JavaBean is an object that is serializable and should have a default constructor with no arguments. Access to the bean's properties is done so using getter and setter methods as shown in Figure 2. /5/

```
public class Person {
    private int     age;
    private String  name;
    ... ..
    public void setAge(int age){
        this.age = age;
    }
    public void setName(String name){
        this.name = name;
    }
    public int getAge(){
        return this.age;
    }
    public String getName(){
        return this.name;
    }
    ... ..
}
```

Figure 2. An example of a class using the JavaBean syntax

2.7 SSM framework

SSM framework is an abbreviation of spring, spring MVC and MyBatis. This can be used to build a variety of enterprise application system.

Spring provides a container to handle a Bean. In configuration file, we can assign particular argument to call constructor of class to initialize object. The core idea of spring is IOC (inversion of control), which is used to increase modularity of the program and make it extensible, which can help the programmer complete initialization of object. /6/

Spring MVC filters user's request. Its servlet (DispatcherServlet) transfer request to HandlerMapping which can match relevant Controller for the request. MyBatis is an encapsulation of JDBC. Its operation is based on SqlSessionFactory. Mapper of MyBatis contains Sql statement for each class. In each operation of a database, SqlSessionFactory will take a SqlSession to execute Sql statement. /7/

2.8 Eclipse

Eclipse is development environment (IDE) for developing applications in languages such as Java , C++, JavaScript, PHP and C. It is the most used IDE for Java applications and is open source released under the terms of the Eclipse Public License. /8/

2.9 Mysql

MySQL is a relational database management system that is free to use as it is an open source project (using the GNU general public license). It was first released in 1995 with the 'SQL' standing for 'structured query language'.

Relational databases store the data in different tables instead of putting all the data in place. The tables are split into rows and columns. Rules are set up between the tables' data fields to define the relationships for example 'one to one', 'one to many', and other constraints 'unique', 'not null' and 'default'. Different types of queries can request specific data from the database for example "SELECT * from customerTable where location = 'Vaasa' " would select all rows that have a value of 'Vaasa' for the location column of a table named 'customerTable. Querying the database for specific data makes the web application a lot more efficient and optimal.

When it was first created, MySQL was intended for small to medium sized project, however as it has progressed in efficiency and capabilities over the past two decades, it can now be used more in large projects depending on the project scope.

Figure 3 shows a basic example of the architecture of a MySQL database and how by using primary keys, tables can be connected. /9/

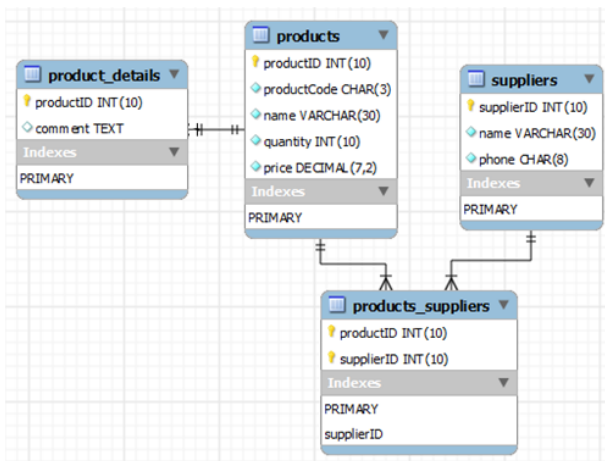


Figure 3. An example of MySQL usage showing a structure of a relational database

3 Analysis

3.1 Requirements

Requirements are divided into the following three types:

1. Normal requirements are must have requirements.
2. Expected requirements are requirements are although not needed, if left out could cause user dissatisfaction with the application
3. Exciting requirements are requirements that would enhance the application and have a low priority.

Table 1. Table of requirements

Requirement	Importance
Use Java technologies	Normal
Products page	Normal
Shopping cart functionality	Normal
Admin view	Normal
Login/register functionality	Normal
User friendly UI	Expected
Easy to understand error handling	Expected

Store user information securely	Expected
Modern designed UI	Exciting

3.1.1 Use Java technologies

Java technologies are flexible, portable, accessible and secure which makes Java the perfect programming language to develop and deploy a secure ecommerce website.

3.1.2 Products page

The products page will show all the products available to buy. Each product will have a name, an image, a description, a price and an option to add the product to the shopping cart. A search bar will allow the user to filter products by typing text into a text field, products with a description containing the input text will be shown in the view.

3.1.3 Shopping cart functionality

The user will be able to add a product to a shopping cart by clicking a 'add to cart' button next to a product on the products page. By clicking on a shopping cart icon, the application will show the shopping cart view. In the shopping cart view, all products in the shopping cart will be shown. Each product in the shopping cart should be able to be deleted or have its quantity changed.

3.1.4 Admin page

An admin page will allow an administrator to update the product list. The admin page will also allow the administrator to view all transactions made on the site.

3.1.5 Login/register functionality

The application will allow user to register to the site with a username and password. Upon logging in, the user's shopping cart is loaded from the database. The user will be able to view their order history.

3.1.6 User friendly UI

The user interface of the application will be designed in such a way that it is easy to navigate for the user. Simple color schemes and fonts will be chosen to keep the UI clear and easy to understand.

3.1.7 Easy to understand error handling

The application will have easy to understand error handling meaning that whenever something goes wrong, the user will be informed so with messages that explain the problem clearly.

3.1.8 Store user information securely

The user information will be stored in a MySQL database table. The user password will be hashed using the SHA-256 algorithm.

4 Design

4.1 Development process

The application development will be divided into three main sections, the database development, the backend development and the frontend development. The frontend will be developed as two separate user interfaces, one for the customers and one for the administrators.

The development process for each view is broken up into the following seven steps:

- Step one: design user interface according to system functions and then develop html page using CSS to style the view.
- Step two: import jar package, integrate three frameworks (Spring, SpringMVC, Mybatis), finish database configuration file.
- Step three: develop controllers in the Web layer according to system functions.
- Step four: develop Service layer according to Web layer.
- Step five: develop Dao layer according to Service layer.
- Step six: show data received from database to html page in JSON format using jquery and ajax.
- Step seven: test the system.

4.2 UI Designs

4.2.1 Customer section

4.2.1.1 Home page

Figure 4 shows the initial design of what the customer home page should look like. Along the top of the page there will be a horizontal list of menu buttons which will take the user to different areas of the website. A large portion of the page will be dedicated to show the products list which will show image and short description of products, as well as a 'add to basket' button. On the right side of the page there will be a list of special offers and a list of popular items.



Figure 4. Mock design of UI for customer home page

4.2.1.2 Registration page

A form will prompt the user to give information such as username, password and email. When the user attempt to complete registration, the form will first be validated. If the validation fails, an alert will give the user information as to why the form validation failed and what the user should do to correct it.

4.2.1.3 Login page

On the login page the user can login by using username and password. After customer successfully login, their username will be shown in the home page, and items in the shopper's own cart will also be shown (if they have items already).

4.2.1.4 Cart page

The cart page is used to show the items in the cart. The customer can see the amount, unit price and total price of each item. The user can submit cart and generate order. The user can also remove items from the cart.

4.2.1.5 Orders page

In the orders page, the customer can see a list of all their own orders and also filter the orders list by some certain terms (item name, unit price, delivered orders, not delivered orders).

4.2.1.6 Profile page

The user can see his/her user information in a form and modify the form. Upon selecting update, the modified user information will be sent and stored in the database.

4.2.2 Admin section

4.2.2.1 Login page

In this page, the admin can login by using username and password. After they successfully login, they will enter the management page. If username or password is incorrect, an error message will be displayed.

4.2.2.2 Item list page

The admin can add, change and delete items on this page. When adding and modifying items, the user can input relevant information and upload image to database. The user can delete items from the item list.

4.2.2.3 Item type page

Item types categorize products to make it easier to search and filter the items. The admin user can view, add, edit and delete item types.

4.2.2.4 Order page

The order page contains three functions: show orders, search orders and show top sales. On the order list page, orders are shown by most recent timestamp first. The admin can handle the state of customers' order. They also can search (by customer' real name, item name and submitted date) and delete orders.

4.3 Database design

Figures 5 through to 10 show the initial ER diagrams for the database.

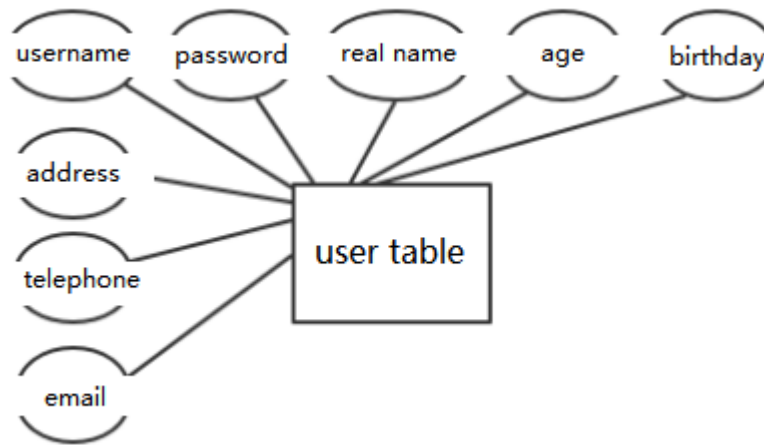


Figure 5. ER diagram of the user table

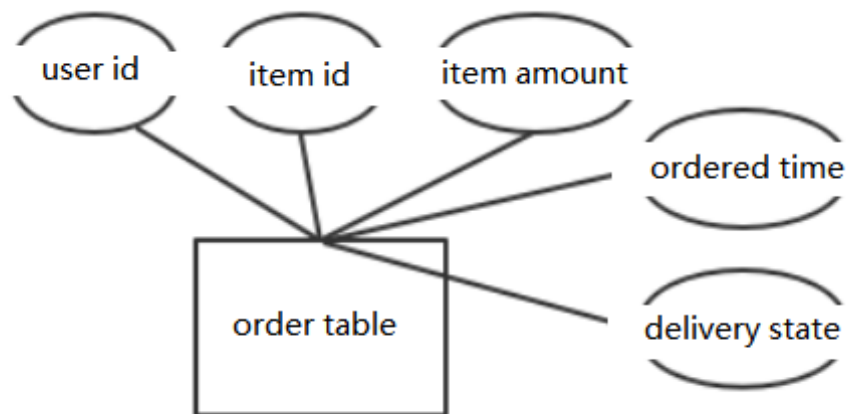


Figure 6. ER diagram of the order table

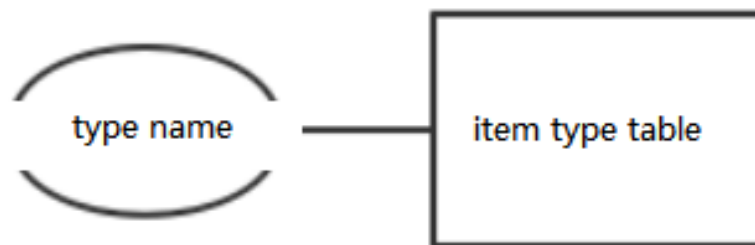


Figure 7. ER diagram of item type table

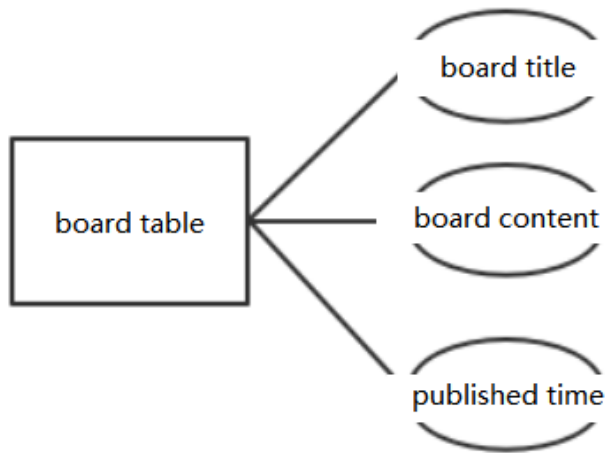


Figure 8. ER diagram of board table

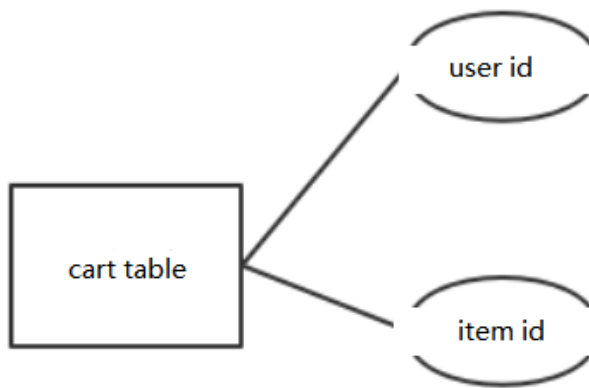


Figure 9. ER diagram of cart table.

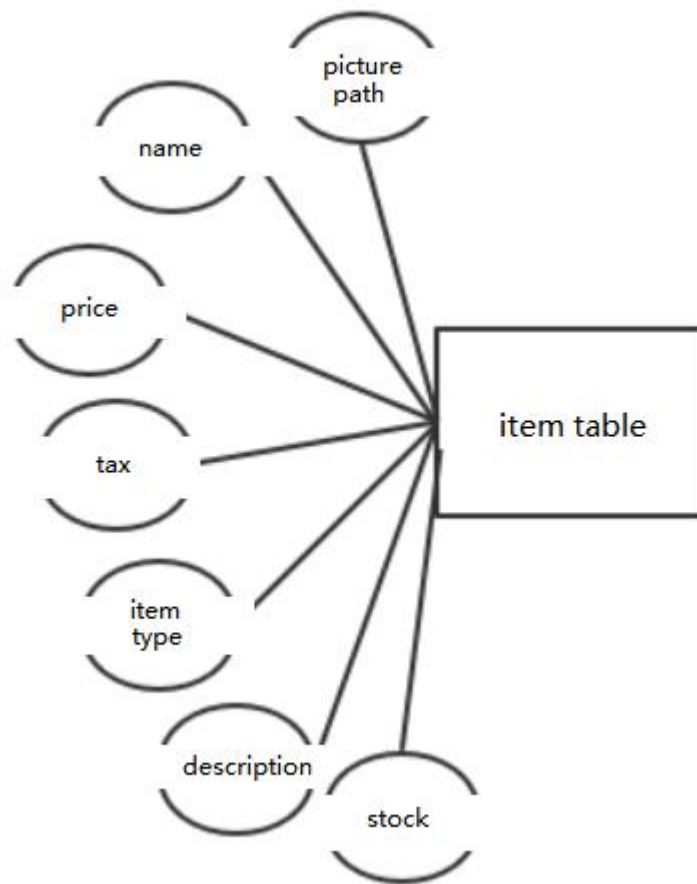


Figure 10. ER diagram of item table.

5 Implementation

5.1 User interface implementation

Figures 11 through to 15 show screenshots of the completed user interface of the customer section. Figures 16 through to 20 show screenshots of the completed user interface of the admin section. HTML, CSS was used to structure and style the pages, whilst Javascript and the JQuery library were used to add content dynamically.

shop

login | register | logout

HOME CART ORDERS PROFILE DELIVERY ABOUT

item name: thinkpad
tax: 200
price: 1111
item component: Screen,Core
item type: compute

item name: imac
tax: 200
price: 2000
item component: mouse
item type: compute

Special offers

- mouse discount 2019-04-12 11:45:21
- new computer 2019-04-12 11:44:53
- watch 2019-04-12 11:45:06
- Welcome 2019-04-13 14:05:10

Cart more>>

item name	price	number
total:	euro	

Top sales

- imac 22
- huawei b5 7
- thinkpad 3
- VicTsing Keyboard 3
- Havit Mouse 2
- ipad 2
- iphone xs max 0

item name: iphone xs max
tax: 600
price: 1500
item component: OLED screen
item type: phone

item name: huawei b5
tax: 15
price: 200
item component: Band
item type: watch

item name: Havit Mouse
tax: 5
price: 100
item component: Wireless Mouse

item name: VicTsing Keyboard
tax: 12
price: 166
item component: Wired

Figure 11. Screenshot of the Home page

login | register | logout

shop

HOME CART ORDERS PROFILE DELIVERY ABOUT

Username:

Password:

login

If you are already a member, please login on the left.

aboutUs | delivery |
Copyright(C)2015-2020

Figure 12. Screenshot of the login page.

hello wine | register | logout

shop

HOME CART ORDERS PROFILE DELIVERY ABOUT

My Cart		
item name	price	amount
imac	2000	1
thinkpad	1111	1
total: 3111euro		
<input type="button" value="submit"/> <input type="button" value="cancel"/>		

about us | delivery |
Copyright(C)2015-2020

Figure 13. Screenshot of the cart page.

hello wine | register | logout

shop

HOME CART ORDERS PROFILE DELIVERY ABOUT

search by item name search

search by unit price search

all my orders not delivered orders delivered orders

item name	customer name	phone number	address	amount	unit price(euro)	total(euro)	order time	delivery state
imac	Henry	13277920317	11#626	2	2000	4000	2019-05-07 17:27:19	not delivered
imac	Henry	13277920317	11#626	2	2000	4000	2019-05-07 17:26:33	not delivered
Havit Mouse	Henry	13277920317	11#626	1	100	100	2019-05-01 21:50:12	not delivered
huawei b5	Henry	13277920317	11#626	2	200	400	2019-05-01 21:50:12	not delivered
imac	Henry	13277920317	11#626	2	2000	4000	2019-05-01 21:50:11	not delivered
thinkpad	Henry	13277920317	11#626	1	1111	1111	2019-05-01 21:50:10	not delivered
huawei b5	Henry	13277920317	11#626	1	200	200	2019-04-13 12:23:00	not delivered

Figure 14. Screenshot of the orders page.

hello wine | register | logout

shop

HOME CART ORDERS PROFILE DELIVERY ABOUT

Please fill in the amendment information.

Username :	wine	*username that you use to log in
Password :	*only English words and numbers.
Real name :	Henry	*please fill in your real name
Age :	21	*please fill in your real age
Address :	vaasa 99	*please fill in your true information.
Phone number :	13277920317	*please fill in your true information.
Email :	634689@QQ.COM	*Please fill in your valid email address so that we can provide you with effective service.
Postal code :	433329	*please fill in your true information.
<input type="submit" value="submit"/>		

Figure 15. Screenshot of the profile page.



Figure 16. Screenshot of the admin login page

Hello, Administrator: admin thank you for landing!

item name	picture	item element	itemType	detailed description	tax	price	number	operation
thinkpad		Screen,Core	compute	Screen Size:15.6 inches Processor:2.6 GHz Intel Core i7 RAM:16 GB DDR4 Hard Drive:512 GB Flash Memory Chipset Brand: NVIDIA	200	1111	9998	update delete
imac		mouse	compute	21.5-Inch (diagonal) 4096-by-2304 Retina 4K display Stunning 5-mm-thin design Quad-core 7th-generation Intel Core i5 Processor Intel Iris Plus Graphics 640 Two Thunderbolt 3 (USB-C) ports 802.11AC Wi-Fi Magic mouse 2	200	2000	9979	update delete
iphone xs max		OLED screen	phone	6.5" OLED screen, HDR and 3D Touch Water Resistant for maximum depth of 2 meters up to 30 mins Dual 12 MP wide-angle and telephoto cameras Face ID, Wireless Charging Branded Case by BOGO Cases	600	1500	10000	update delete
huawei b5		Band	watch	Band Length: 145 - 240mm.Replacement watch band for Huawei Talkband B5 The strap buckle is stainless steel flip lock diver clasp for additional protection.more safely.All mesh bracelet passed pulling force test,professionally made and constructed,quality assured, more soft and comfortable,also can easy to adjust the length to fit your wrist, is a very good cost-effective strap	15	200	9995	update delete
Havit Mouse		Wireless Mouse	mouse	MINI COMPACT SIZE - 3.89*2.55*1.44" Mouse body, especially design for small right hand. Thanks to its extra-portability pocket-sized, you can take the wireless mouse wherever with your laptop. 2.4GHz WIRELESS - A more secure and stable connection with an increased working distance up to	5	100	9999	update delete

Figure 17. Screenshot of the item management page

Hello, Administrator: admin thank you for landing!

Item Management

- Add a new item
- Item list
- Item Type Management**
- Add a new item type
- Item type list
- Message Management**
- Add a new message
- Message list
- Order Management**
- Order list
- Search order
- Top sales
- Admin Management**
- Exit

Item Type Information

Type ID	Type Name	operation	
1	pad	update	delete
2	compute	update	delete
3	telephone	update	delete
4	phone	update	delete
5	mouse	update	delete

Total: 2 Pages, [[Previous Page](#)][No.1Page](#)[Next Page](#)

Figure 18. Screenshot of the item type page.

for landing!

Item Management

- Add a new item
- Item list
- Item Type Management**
- Add a new item type
- Item type list
- Message Management**
- Add a new message
- Message list
- Order Management**
- Order list
- Search order
- Top sales
- Admin Management**
- Exit

Message Information

Title	Content	Release Time		
mouse discount	Save 30% on all mouses!	2019-04-12 11:45:21	update	delete
new computer	New computer model arrived!	2019-04-12 11:44:53	update	delete
watch	Save 40% on watch!	2019-04-12 11:45:06	update	delete
Welcome	Hello everyone! Hope you enjoy our shop!	2019-04-13 14:05:10	update	delete

Total: 1 Pages, [[Previous Page](#)][No.1Page](#)[Next Page](#)

Figure 19. Screenshot of the notice management page

Hello, Administrator: admin thank you for landing!

Item Management

- Add a new item
- Item list
- Item Type Management**
- Add a new item type
- Item type list
- Message Management**
- Add a new message
- Message list
- Order Management**
- Order list
- Search order
- Top sales
- Admin Management**
- Exit

order Information List

UserID	RealName	PhoneNumber	Address	ItemName	Number	UnitPrice(yuan)	TotalPrice(yuan)	OrderTime	Deliver	Confirm	
13	Jackson	15678941110	10#101	ViCtSIng Keyboard	1	166	166	2019-04-12 14:47:50	not delivered	confirm	concel
13	Jackson	15678941110	10#101	ViCtSIng Keyboard	1	166	166	2019-04-12 15:25:58	not delivered	confirm	concel
13	Jackson	15678941110	10#101	ipad	1	1500	1500	2019-04-12 15:26:17	delivered	confirm	concel
13	Jackson	15678941110	10#101	imac	1	2000	2000	2019-04-12 15:41:59	not delivered	confirm	concel
13	Jackson	15678941110	10#101	huawei b5	1	200	200	2019-04-12 15:41:59	not delivered	confirm	concel

Total: 4 Pages, [[Previous Page](#)][No.1Page](#)[Next Page](#)

Figure 20. Screenshot of the order management page

5.2 Functionality

5.2.1 Outline

According to the requirements of customer and admin and the tables in the database, six modules were created for the backend. These six modules have similar structures to each other. The following list, lists the differences between them.

1) User module

The user can login by username and password. If they successfully login, the system will return the user's information, otherwise it will return null.

Requirement: user login

URL: /user/login

Parameters: username, password

Returned values: success: user information, failure: null

The frontend can acquire current user's information whenever it is needed.

Requirement: acquire current user information

URL: /user/getCurrentUser

Parameters: no parameters

Returned values: success: user information, failure: null

When user log out, all the information related will be removed at the backend.

Requirement: user log out

URL: /user/quit

Parameters: no parameters

Returned values: success: "ok", failure: null

2) Cart module

A customer can submit their cart to generate order. After that, items in the cart will be removed and the order will be shown in order table.

Requirement: user buy items

URL: /cart/buy

Parameters: ids of items

Returned values: success: "ok", failure: null

3) Order module

A customer and the admin can search order by some terms. There are five optional parameters.

Requirement: search order

URL: /order/queryList

Parameters: realName, itemName, date, page, pageSize

Returned values: success: order list, failure: null

4) Item module

A customer and the admin can see the top sales.

Requirement: show popular items

URL: /item/topList

Parameters: page, pageSize

Returned values: success: item list with sale amount, failure: null

A user can acquire an item information by its id.

Requirement: get item information

URL: /item/getItemById

Parameters: item id

Returned values: success: item information, failure: null

5) Item type module

Functions here are simple. Admin can add, search, update and delete item type.

6) Bulletin board module

A user can acquire details (title, content and date) of a message in bulletin board by its id.

Requirement: get message detail

URL: /board/getBoardById

Parameters: message id

Returned values: success: message information, failure: null

5.2.2 Functionality example

This sub chapter uses the Top Sales list on the home page as an example to show how the functionality of the application works starting from the frontend requesting the data and finishing with the frontend receiving the data and displaying it on the user interface. Figures 21 through to 25 show code snippets to demonstrate this functionality.

Figure 21 shows a code snippet of the Javascript `initTopList` function, which is called on page load of the home page. This function uses a JQuery Ajax request function which makes a get request to the item controller. If the data is successfully returned

then a for each loop is used to append the items into a list in the HTML using the HTML elements ID, 'mytop'.

```
function initTopList(){
    $.ajax({
        type : "get",
        url : "/item/topList",
        dataType : "json",
        data:{
            page:1,
            pageSize:10
        },
        success : function(data) {
            $.each(data.list, function (i, item) {
                $("#mytop").append("<li><b>" + item.itemName + "</b>" + "<font");
            });
        },
        error : function() {
            alert("Sorry, an error occurred");
        }
    });
}
initTopList();
```

Figure 21. Code snippet of the Javascript initTopList function.

Figure 22 shows a code snippet of the ItemController class which contains a method topList which returns the topList method from the itemService.

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.multipart.MultipartFile;

import com.github.pagehelper.PageInfo;
import com.shop.dto.PageDto;
import com.shop.pojo.Item;
import com.shop.service.ItemService;

@RestController
@RequestMapping("/item")
public class ItemController {

    @Autowired
    private ItemService itemService;

    @RequestMapping("/topList")
    public PageInfo<Item> topList(PageDto pageDto) {
        return itemService.topList(pageDto);
    }
}

```

Figure 22. Code snippet of ItemController showing topList method.

```

package com.shop.service;

import org.springframework.web.multipart.MultipartFile;

import com.github.pagehelper.PageInfo;
import com.shop.dto.PageDto;
import com.shop.pojo.Item;

public interface ItemService {

    public PageInfo<Item> topList(PageDto pageDto);
}

```

Figure 23. Code snippet of the ItemService interface.

Figure 24 shows the topList method in the ItemServiceImpl which returns item list retrieved using the ItemMapper class. The plugins com.github.pagehelper.PageHelper

and `com.github.pagehelper.PageInfo` are used to determine how large the page is and how many items will be displayed on each page. The `ItemMapper` like all mapper files are auto generated by the `generatorConfig.xml` file the database table schema was inputted.

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;

import com.github.pagehelper.PageHelper;
import com.github.pagehelper.PageInfo;
import com.shop.dao.ItemMapper;
import com.shop.dto.PageDto;
import com.shop.pojo.Item;
import com.shop.pojo.ItemExample;
import com.shop.service.ItemService;
import com.shop.utils.FileUtil;
@Service
public class ItemServiceImpl implements ItemService {

    @Autowired
    private ItemMapper itemMapper;

    public PageInfo<Item> topList(PageDto pageDto) {

        PageHelper.startPage(pageDto.getPage(), pageDto.getPageSize());
        ItemExample itemExample = new ItemExample();
        itemExample.setOrderByClause("itemMarkNumber DESC");
        List<Item> items = itemMapper.selectByExample(itemExample);
        PageInfo<Item> pageInfo = new PageInfo<Item>(items);
        return pageInfo;
    }
}
```

Figure 24. Code snippet of the `ItemServiceImpl` class

Figure 25 shows a code snippet from the `itemMapper.xml` file. The code snippet contains the SQL query for the item list.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper
<mapper namespace="com.shop.dao.ItemMapper" >
  <resultMap id="BaseResultMap" type="com.shop.pojo.Item" >
    <id column="id" property="id" jdbcType="INTEGER" />
    <result column="itemName" property="itemName" jdbcType="VARCHAR" />
    <result column="itemMarkPrice" property="itemMarkPrice" jdbcType="DOUBLE" />
    <result column="itemPriPrice" property="itemPriPrice" jdbcType="DOUBLE" />
    <result column="itemType" property="itemType" jdbcType="INTEGER" />
    <result column="itemElement" property="itemElement" jdbcType="VARCHAR" />
    <result column="itemMarkNumber" property="itemMarkNumber" jdbcType="INTEGER" />
    <result column="itemPriNumber" property="itemPriNumber" jdbcType="INTEGER" />
    <result column="itemDes" property="itemDes" jdbcType="VARCHAR" />
    <result column="iconPath" property="iconPath" jdbcType="VARCHAR" />
  </resultMap>

  <select id="selectByExample" resultMap="BaseResultMap" parameterType="com.shop.pojo.ItemExample" >
    select
    <if test="distinct" >
      distinct
    </if>
    <include refid="Base_Column_List" />
    from t_item
    <if test="_parameter != null" >
      <include refid="Example_Where_Clause" />
    </if>
    <if test="orderByClause != null" >
      order by ${orderByClause}
  </select>

```

Figure 25. Code snippet of itemMapper.xml containing SQL query

5.3 Database implementation

Tables 2 through to table 7 show the implemented SQL table structures.

Table 2 shows the implementation of the user table.

Table 2. Implementation of the user table

Field name	Field type	Restraint condition	Explanatory note
id	INT	primary key	user id
userName	VARCHAR(50)	not null	username
password	VARCHAR(50)	not null	password

realName	VARCHAR(50)	not null	customer name
age	INT	not null	age
address	VARCHAR(50)	not null	address
phone	VARCHAR(50)	not null	phone number
email	VARCHAR(50)	not null	email

Table 3 shows the cart, which relates with the user table and the item table, so it contains user id and item id as foreign key.

Table 3. Implementation of cart table.

Field name	Field type	Restraint condition	Explanatory note
Id	INT	primary key	car id
userId	VARCHAR(20)	not null	user id
itemId	VARCHAR(20)	not null	item id

Table 4 shows the order table which relates with the user table and the item table so it has the user id and item id as foreign key. ItemNum is calculated by the user id and item id in the cart table. The orderTime is when order is submitted is saved as datetime type. OrderIsSub is a mark which indicates whether the order is delivered or not.

Table 4. Order table implementation.

Field name	Field type	Restraint condition	Explanatory note
id	INT	primary key	Order id
userId	INT	not null	user id
itemId	INT	not null	item id
itemNum	INT	not null	item amount
orderTime	datetime	not null	submitted time
orderIsSub	INT	not null	delivered state

Table 5 shows the Item table which has id as primary key but no foreign key.

Table 5. Item table implementation.

Field name	Field type	Restraint condition	Explanatory note
id	INT	primary key	item id
itemName	VARCHAR(50)	not null	item name
itemMarkPrice	INT	not null	tax
itemPriPrice	INT	not null	item price

itemType	INT	not null	item type
itemElement	VARCHAR(100)	not null	item component
itemMarkNumber	INT	not null	sold amount
itemPriNumber	INT	not null	stock number
itemDes	Text	not null	item description
iconPath	VARCHAR(100)	not null	picture path

Table 6 shows the Item type table which only has two fields: id as primary key and typeName. The id here is a foreign key in the item table.

Table 6. Item type table implementation.

Field name	Field type	Restraint condition	Explanatory note
Id	INT	primary key	item type id
typeName	VARCHAR(20)	not null	item type name

Table 7 shows the Board table which is an independent table meaning it has no relation with the other tables.

Table 7. Implementation of the board table

Field name	Field type	Restraint condition	Explanatory note
id	INT	primary key	board id
boardTitle	VARCHAR(20)	not null	board title
boardMsg	VARCHAR(20)	not null	board content
boardTime	datetime	not null	published time

6 Testing

The finished web application was tested using the black box testing technique which is a way of testing the functionality of an application without accessing into its internal structure. Each part of the application was tested to see if the functionality works and if all the requirements were met. First all the controllers were tested using correct parameters and then tested again using incorrect parameters.

Use case testing was used for all functionality of the application, for example, to test whether the system can show a notice about success after customer registration.

7 Conclusion

The completed ecommerce web application allows the user to register, log in and log out of the application. When the user registers to the website, the information inputted is first validated. The user can add products to the shopping cart and order the products. The user's orders are stored in the database and the user can view and filter their order history at any time. The site admin can log in to the admin section of the application to add, edit and delete items as well as add, edit and delete special offers. The application was developed using Java technologies, MySQL, HTML, JavaScript and CSS. Although the SSM framework added extra layers of abstraction to the code, it made the development of the application easier to manage and easier to add new features after the initial implementation.

8 References

/1/ HTML – Last access 10/03/2019

https://www.w3schools.com/html/html_intro.asp

/2/ CSS – Last access 10/03/2019

https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/How_CSS_works

/3/ Java – Last access 10/03/2019

https://www.java.com/en/download/faq/whatis_java.xml/2/Java_spring_MVC_Last_access_10/03/2019

/4/ J2EE – Last access 10/03/2019

<http://www.informit.com/articles/article.aspx?p=26660>

/5/ Java beans – Last access 10/03/2019

<https://www.techopedia.com/definition/7865/javabeans>

/6/ Java spring – Last access 10/03/2019

<https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html>

/7/ MyBatis – Last access 10/03/2019

<http://www.mybatis.org/spring/>

/8/ Eclipse – Last access 10/03/2019

[https://en.wikipedia.org/wiki/Eclipse_\(software\)](https://en.wikipedia.org/wiki/Eclipse_(software))

/9/ MySQL – Last access 10/03/2019

<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>