

Shopping simulating system

Software Design Specification



November 13, 2016

Huizhe Wang

1. **Introduction**
   1. **System Overview**

This Shopping Simulating System serves as an aid for sellers to collect the customers’ orders. It is free even though the customers make orders.

In the System, the product information will be presented in the customers’ screen by sellers. Customers could search these products by departments, sort them by price, size or arrival time, and then make their orders. Moreover, customers could give sellers and other customers advise of products. For sellers, the main functions of the Shopping Simulating System are uploading the product information, managing the orders from customers and give customers feedback.

The benefits of the Shopping Simulating System will be far-reaching. The records of customers’ orders could be analyzed by sellers and help them grasp the majority customer demand. For customers, they could to be hip to the latest styles without pay money.

* 1. **Definitions, Abbreviation and References**

*Definition and Abbreviation*

1. SSS: Shopping Simulating System – the software application that is being discussed in this design specification.
2. MVC: Model-View-Controller – a software design pattern for implementing user interfaces on computers.
3. DAO: Data Access Object – an object that provides an abstract interface to some type of database or other persistence mechanism.
4. WWW: World Wide Web: a collection of hyper text documents accessed via the Internet.
5. JEE: Java Enterprise Edition: a widely used platform for server programming in the Java programming language.
6. Java: a system for developing cross-platform application software.

*References (3rd party Frameworks)*

1. Eclipse: <https://eclipse.org/downloads/>
2. Spring: <http://www.springsource.org/about>
3. Spring Security: <http://projects.spring.io/spring-security/>
4. MySQL: <http://www.mysql.com/>
5. **Design Considerations**
   1. **Assumptions**

The operational environment required for the SSS is a Spring Source server such as Tomcat Server. The database of the SSS client is MySQL. The SSS is independent and not a component of a larger system.

* 1. **Constraints**

Certain data constraints should be in effect since excessive data transmission with photos would reduce operating speed.

* 1. **System Environment**

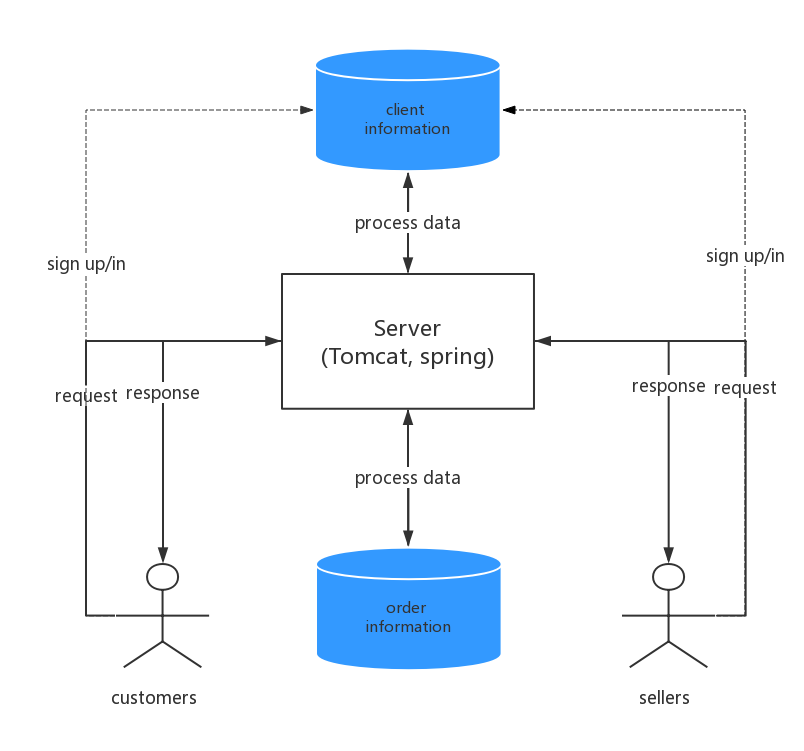
The software that the Shopping Simulating System server will run on is the SpringSource Java Application Server.

Integral to the operation of the Shopping Simulating System are certain services that manage data and business logic. These services that the SSS interacts with are some fashion web, likes [BAZZAR](http://www.harpersbazaar.com/), [VOGUE](http://www.vogue.com/magazine/), [ELLE](http://www.elle.com/) and so on.

* 1. **Design Methodology**

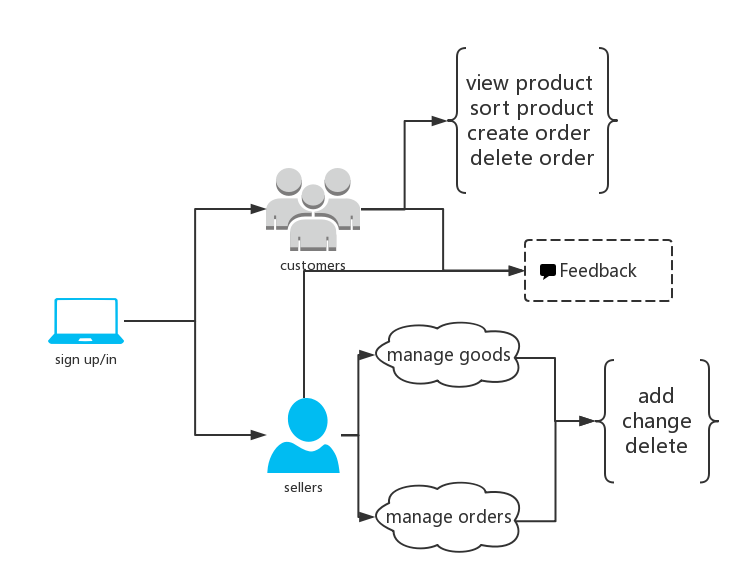
The design methodology being employed here is the UML object-oriented technique insofar as necessary since large parts of the SSS functionality are implemented by 3rd party services that have been designed and implemented already.

1. **Server (Spring Web App)**

****

1. **Client (Customers and Sellers)**

Shopping Simulating System for customers and sellers:



use case diagram

* 1. **Sign up/in**

1. **Introduction:** the SSS allows the customers and sellers to access the different web pages. The database which storing users’ information is different from the product one. When users sign up, the database will store their usernames and passwords. When users sign in, the database will judge whether their usernames and passwords match or not. Moreover, users could reset when they sign up/in, and change their passwords.
2. **Inputs:** Users will have authenticated to the SSS server and send their requests.
3. **Outputs:** The server will deal with their request and send them to the right pages. Customers has no right to enter into sellers’ pages, and vice versa.
   1. **Customers**
      1. **view/sort products**
4. **Introduction:** The SSS allows customers view details of products and sorts products with department, price, or arrival time.
5. **Inputs:** user input is available if customers want to search the products they want to see. However, in most case, they would choose to select the options given by sellers.
6. **Outputs:** Visible output on the screen will consist of the details of the products selected along with any additional requests, such as “the price from 50 dollars to 200 dollars”.
   * 1. **create/delete orders**
7. **Introduction:** The SSS allows the customers work with his or her own orders. An appropriate form will be presented to the customers so that he or she can create orders. Moreover, an appropriate option will also be presented so that customers want to delete their orders. Deleting a calendar event will delete the information stored in the database.
8. **Inputs:** No user input is required other than the selection of the product information given by sellers, such as the size, order number and color of clothes.
9. **Outputs:** Only visible output on the screen would be a confirmation message that the orders have been created or deleted.
   1. **Sellers**
      1. **manage products**
10. **Introduction:** The SSS allows the sellers to upload, add, change and delete the product information which will display on the customers’ screen.
11. **Inputs:** User input is available if sellers want to add some information the web developer doesn’t give. However, in most case, they would choose to select the options given by the manager, such as the arrival time, price, color and department.
12. **Outputs:** The uploaded, added and updated information would be displayed on the screen. Only visible output on the screen would be a confirmation message that some products have been deleted.
    * 1. **manage orders**
13. **Introduction:** The SSS allows the sellers to view customers’ orders directly. Sellers could sort the orders by time or number. Moreover, sellers could delete customers’ order with sending an email to that customer.
14. **Inputs:** The user must provide the information that he or she wants to delete customers’ orders.
15. **Outputs:** With sending an email to customers, only visible output on the screen would be a confirmation message that the customers’ orders have been deleted.
    1. **Feedback**
16. **Introduction:** The SSS allows all the users – customers and sellers – to upload, edit and delete their own words. They can view all the users’ messages as long as they have signed up/in.
17. **Inputs:** Users input is necessary. An appropriate form will be presented to the users so that he or she could leave their words in the message, which would be viewed by all users.
18. **Outputs:** The whole feedback page would be displayed on the screed with the newest messages.
19. **Logical Data Requirements**

This application has two databases. One is for the user information, such as first name, last name, email, username and password. The other is for product information, such as how many do customers want to buy it, the feedback with the product and so on.

1. **User Interface Design (Support Information)**

No additional supporting information at this time.