

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Design and Realization of Car Rental Management System Based on AJAX+SSH

Zhang Li

College of Computer Engineering, Jiangsu University of Technology,
Changzhou, 213001, Jiangsu, China

Abstract: With the high speed development of car rental market, the traditional manual rental management is not enough for the various business information processes. Therefore, this study proposed an enterprise-class development program of car rental management system based on AJAX+SSH and this program will be applied in this development of the car rental management system. This article introduced the accomplishment of the functional module and the design of the whole structure. Practice proved that this program can not only greatly increase the developing efficiency of this system but also reduce the difficulty; meanwhile, it can raise the response efficiency of Web application system and the effect of user experience.

Key words: Car rental management system, MVC pattern, AJAX framework, SSH framework, class diagram, use case diagram

INTRODUCTION

With the change of people's vision and working way, more and more families and persons don't need to own a car in long term but they turn to a more convenient way to use cars. Rent cars can not only save much money and people don't need to pay for road maintenance, insurance and maintenance costs but also they don't need to pay for garage and parking. Because people can rent a car by hour or by day, even they can require for a driver. Because of these, car rental industry is developing rapidly. However, there are also some shortages during business operation, for example, low efficiency, disordered management, more mistakes and so on. Hence, in reality, it is very essential to adopt computer information management in car rental industry and it is also necessary to upgrade and improve the management level. Meanwhile, it is significant to increase the service efficiency. Thus, it is an important way and main trend to apply internet in car rental industry.

This article is based on AJAX+Struts (Jiang *et al.*, 2009; Zhang and Zhang, 2010)+Spring+Hibernate of MVC (Gamma *et al.*, 2009) and it discussed the design of car rental management system based on WEB2.0. Customers can check if there are some cars suitable for them to rent without going to the leasing stores. Meanwhile, for the leasing companies, it changed from one fixed placed to several different places, it is much more convenient for the customers, this can save money and time for both leasing companies and customers.

REQUIREMENTS OVERVIEW

The traditional car rental industry relies on the artificial operation; it is hard to deal with the large number data processing. Meanwhile, the price of oil is increasing and the car parking is more difficult, these problems make the customer feel that they can buy a car but they cannot afford to keep a car. This trend requires a new car rental way that is car rental system under internet. It can relieve the pressure during peak time and it can offer a convenient service to customers. According to the analysis of manual processes, this article divides the car rental management system (Fig. 1) to seven models: it mainly includes type management (car type and car insurance type), car management, insurance company management and vendors' management:

- **Car type management:** System administrator can operate on car type (car name and detailed description)
- **Insurance type management:** System administrator can operate on car insurance type (the name of insurance and the detailed description)
- **Vendors' management:** System administrator can inquire on car vendors (the name of vendors) and can delete and check the information of sold cars
- **Insurance company management:** System administrator can add and modify (to the selected record) information of the inquired insurance company (by company name), delete the information of insurance company and check the insured car information

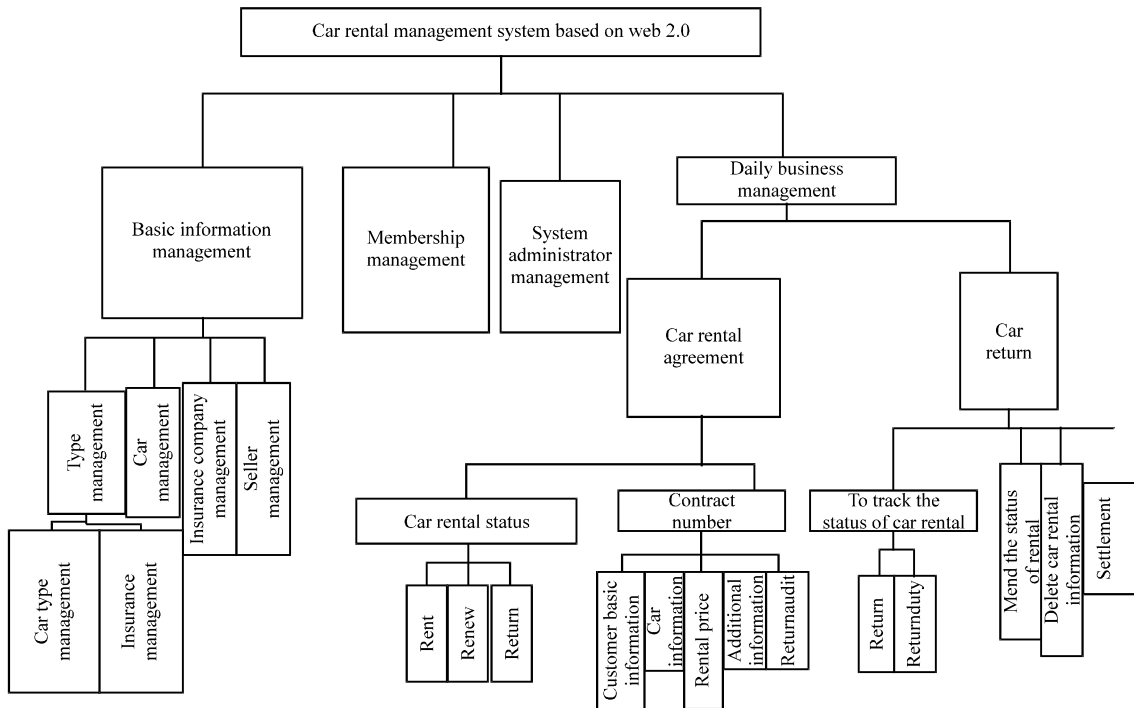


Fig. 1: Car rental management system based on web2.0

- **Car management:** System administrator can seek the related information of this car and the rental information (car plate number) of all the cars in this company, add (basic information, insurance information, rental information and additional description), mend and delete some information. It mainly includes: Customer information management, members information management, members type information management
- **Members information management:** System administrator can look for the members (by name) and look for the customers' (drivers) information and some related driving information and mend some information (selected: Basic information, driving license, collateral information and membership information)
- **Membership type information:** System administrator can add membership type (type name and discount) and mend (selected) and delete some information

USE CASE MODELING OF CRMS

There is a conclusion according to the analysis in chapter 1 that firstly, this system is facing to most customers who have the membership with the car rental companies. Secondly, it is facing to the service offers who are the workers in the car rental companies. They are

charging for the daily rental business. Lastly, it is facing to the system maintainer and the system administrator. Therefore, there are three kinds of participants in this system. However, at present, requirements' modeling is using Microsoft Visio to modeling. The advantage of using this is to seamlessly integrate the newest method of software engineering to UML. It can easily make the modeling under this way. The modeling is showing in Fig. 2.

STATIC STRUCTURE MODEL OF CRMS

Class diagram (Metsker and Wake, 2009) is one of the most important UML, it respectively described the static structure and relationship of class and class, class and subclass, it used to describe the structured design of system. Figure 3 is the simplified model of a class diagram of the car rental management system. The extension mechanism of class diagram in UML divides the system into nine prototype classes. The detailed description is in the following. the relationship between user type and user is one to many. The user and the company is many-to-one. The car and the car type is one to many. That means a car can only be one car type but one car type can have many cars. The car and the insurance serve is many-to-one. It means a car insurance serve can serve many cars but one car can only get one car insurance.

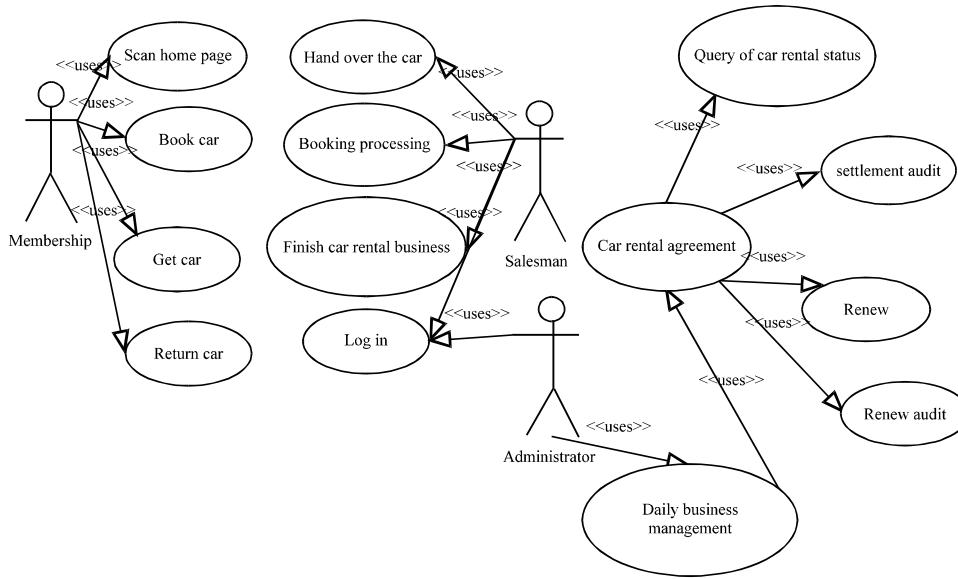


Fig. 2: Use case modeling of CRMS

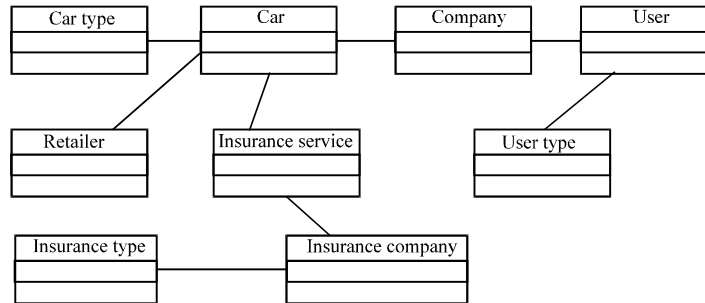


Fig. 3: Static structure model of CRMS

Because the insurance type can be added in insurance service, insurance company and insurance type is many-to-one. One insurance type can be sold to many insurance companies while one insurance company can sell many kinds of insurance. Insurance type and insurance service is many-to-many. It means one insurance type can be added to many insurance services while one insurance service can be added in many insurance types.

INTRODUCTION OF AJAX+MVC+SSH

AJAX is Asynchronous JavaScript and XML (asynchronous JavaScript and XML), AJAX (Zhang and Zhang, 2009, 2009) is not abbreviations but is created by Jesse James Gaiett, is a web development technology of creating interactive web applications. MVC is a designed model, it forcibly separates the input, processing and output. Therefore the application program

is divided to three main parts: model, view and controller. They finished their task separately. Struts is a structure based on MVC. Hibernate is a mapping framework based on object/relational database, it has two functions, one is to manage the mapping structure from Java to database, another is that it has the strongest inquiring language (Hibernate Query Language, HQL), which can to a large extent reduce the data processing time by using SQL and JDBC and it decreased the development difficulty. Core container offers the basic functions of Spring structure, the main components is Bean Factory which use IOC to separates configuration and dependency specification and the actual application code.

DESIGN OF SYSTEM PROCESS

According to the design of MVC, AJAX+SSH based on CRMS divided the system into presentation layer (AJAX+JSP), controller layer (ActionServlet,

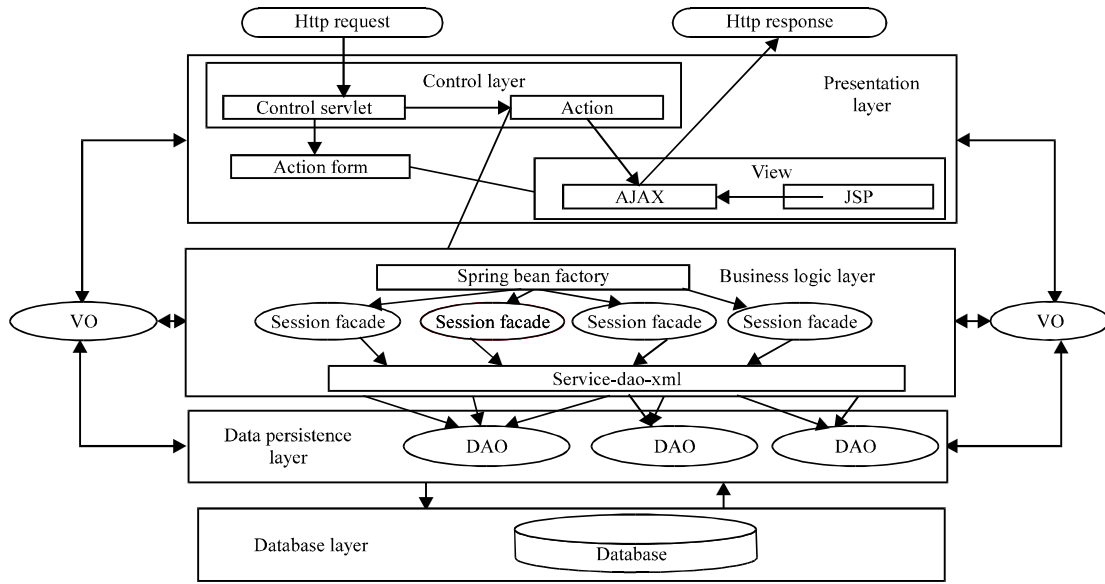


Fig. 4: Integration of CRMS

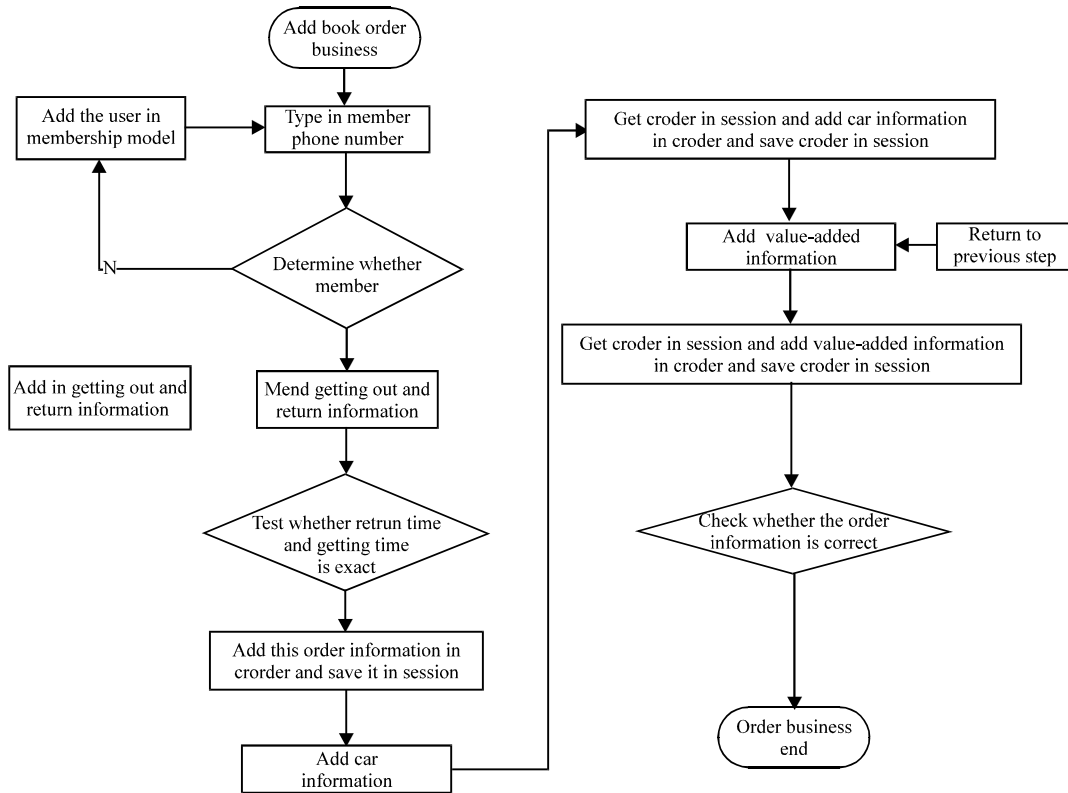


Fig. 5: Order booking process

RequestProcessor and Action), business logic layer (Spring Bean Factory), data access layer. The aim of each layer is clear, the advantage is easy to maintain. It is showing in Fig. 4.

The first step, integrate presentation layer. The presentation layer is composed by AJAX+Struts and JSP. View is constructed by JSP. Struts have a set of tag libraries which can simplify the codes on JSP. Meanwhile,

AJAX is one of the core technologies of WEB2.0. It is alike a middle tier between user and server which can make the users' operation and server's response asynchronous. Therefore, it transfers some work of server to client side to reduce the burden of server and broadband. It can save the JSP space and decrease the costs of broadband and increase the developing efficiency.

The second step, integrate the control layer. There are four tasks that the control layer has to finish. One is to accept the customer's request. Two is to choose the suitable model components to operate the corresponding business logic according to the customers' request. Three is to achieve the result from business logic. Four is to choose suitable view components and feed them back to customers according to the result. These four tasks are controlled by ActionServlet and RequestProcessor.

The third step is to use Spring to integrate business logic layer. The specific detail is to do the DAO management in Spring. Here to use Façade to do the integrating work. The advantages of using Façade is that it can reduce the coupling of presentation layer and business logic layer. Meanwhile, it can save information for Action in control layer which can increase the service of http request. Moreover, it means that it can prevent the burden of DAO.

The last step is to use Hibernate+DAO (Zhang, 2010) to integrate data access layer. From 3.1, the function of DAO is to pack the process of accessing database and that is to finish the CRUD of VO. Through Hibernate, it can finish the operation which has no relation to database and can hide some details of data source. Meanwhile, it can simplify the operation of adding and deleting database. It can guarantee that under different system storage pattern, it won't affect the business logic layer. Therefore, to use Hibernate+DAO, there will not be too many codes modification.

REALIZATION OF CRMS OF AJAX+SSH BASED ON MVC

To CRMS, all the structures are similar. Take booking car for example to show how AJAX+SSH is realized in CRMS. The booking of the car is the most complex business in all operations (Fig. 5). This can be divided to four JSP, they are: add order, get and return the car, car information, basic insurance and value-added service information. The last JSP is order confirmation page (Zhang and Zhang, 2009). The specific details are as following: after submitting the order, system will add the order information in session to database and the order will be feedback to clients. For saving the order information,

the order information for each time will be added in session, in the next JSP, if need the information of order, the order information can be taken out and be used in business operation. In the former JSP, if there are some mistakes, in the next JSP, if it needs to go back to the former JSP, the information is still there.

REALIZATION IN PRESENTATION LAYER

The page design needs Dreamweaver and Photoshop, each JSP needs combine to Struts and AJAX to be constructed. The advantage is that the processor can pay more attention to the business and the art design can beautify the view. The detailed steps are as following: first, according to diagram 5, to construct every JSP: SearchCat.jsp (Zhang and Zhang, 2010) which can search the car information; catInfoMaint.jsp which can maintain the rental car information, it can show all the information of cars and here the CRUD can be done. Second, describe and definite OrderForm. Because OrderForm inherits org.apache.struts.action.ActionForm. The main function of this class is to gather and show the data. Gathering the data is to supply data to control layer, the data presentation is showing to customers by AJAX and Struts.

TECHNIQUE CHARACTERISTICS

This system adopted B/S framework, the advantage is that the client can use browser directly to visit the system and database resource, therefore, the needed information can be got. It to a large extent decreases the application cost of the system. In terms of view, it adopted jQuery+jQuery EasyUI, to make the client request and the server response be asynchronously exchanged. It saved the band width greatly and shortened the time. It makes the system fast, flexible, maintainable and extensible and it solved the problem that huge database can share the information in long distance.

REALIZATION OF BUSINESS LOGIC

After realization of presentation, from diagram 5, when enter the car rental page, the business of back stage will trigger codes in Action to analysis the whole realization process. The detailed process is: first, read the files in Struts-config.xml; second, according to OrderForm in Struts-config.xml to find the query in OrderAction of Action Mapping. If the query was not found, the request page will turn to 404, otherwise, the query will be done. The specific query conditions are: order number, user name, member phone number, car number, time out, time

```
public String query(){
    crorderList = crorderService.searchCROrder(
    .....
    crorder!=null?crorder.getReturnTime():null,
    minLeaseTerm, maxLeaseTerm, minAllRent, maxAllRent, pageNum);
    maxPageNum = 1;
    if(pageNum<=0){pageNum=1; }
    int count = crorderService.searchCROrderCount(
    minLeaseTerm, maxLeaseTerm, minAllRent, maxAllRent);
    if(count%10!=0){
        maxPageNum = count/10+1;
    }else{maxPageNum = count/10;}
        memberMap = memberService.getMemberMap();
    return result;
}
```

Fig. 6: Query codes of car rental page

back, lease online, lease offline, rent on line, rent offline. These can be collocated and combined freely. The codes are as following in Fig. 6. The crorderService is the object in business logic. crorderService includes all the processing methods of business logic for car order. Then use searchCROrder () and searchCROrderCount () in crorderService to find CrorderDao in spring-dao.xml .And then to adjust the HQL by SearchCROrder () and Count () in CrorderDao and here HQL is the codes of accessing data layer. Finally save these results in session and also turnn the results to orderSubmit.jsp and present the successful results.

CONCLUSION

This system used AJAX+SSH to realize the car rental management system. It can totally realize the functions of

query, booking, manage and maintain. The advantage of this system is that it makes it real to conveniently and quickly rent a car. In this system it is easy to get car information and book a car and it disperses the pressure of renting a car and scattered the rental crowds. Meanwhile, the layer in system is very clear and simple and the coupling in each layer is very low, the writing of every code is very simple, therefore the system development efficiency is high and it also raised the reliability and maintainability.

REFERENCE

- Gamma, E., R. Helm, R. Johnson and J. Vlissides, 2009. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley/Pearson, USA.
- Jiang, H.C., Y.X. Yin and X.J. Ban, 2009. Struts framework application based on SOA. Comput. Eng., 35: 21-23.
- Metsker, S.J. and W.C. Wake, 2009. Design Patterns in Java. Addison Wesley, USA.
- Zhang, L. and W.X. Zhang, 2010. Design and realization persistence framework of tourism e-business system. J. Wuhan Univ. Technol., 32: 113-117.
- Zhang, L. and W.X. Zhang, 2009. Design and implementation of fixed assets management system based on JavaEE. Comput. Eng. Design, 30: 3797-3800.
- Zhang, L., 2010. Application of MVC pattern in data middleware. Comput. Eng., 36: 70-72.