YITONG Express System Design

Group 9. Dec.19.2015

Content

Introduction

Purpose of the document

using Unified Modeling Language (UML), based on object-oriented system analysis and design method to realize Management System for Logistics Business.

Through the demand analysis, we make filter, integration and optimization of traditional logistics systems. We are aimed at designing a system which is personalized, practical and available.

on the disadvantages of traditional Management System for Logistics Business, we will improve the system stability, robustness, availability, scalability and operability. In order to provide a better user experience.

System overview

Preliminary Work

In software requirements specification, we have already analysed and compared exiting system, analysed the requirements. We decide the major function by different modules. In this system, users are divided into three roles: customers, the postman and deliverer. We analysis system from following scenarios step by step: making orders, the distribution of goods, delivery, payment etc. We also provide use case diagram and use case specification to describe the services provided by the system in details, which shows relationship between the participants and the use case, use case and other use case.

Present Work

In this work, we preliminary implementing system architecture and modeling the functions. In addition to use class diagrams to describe the static structure of the system, we also use sequence diagram and communication diagram to describe the dynamic structure of system in order to clear the interaction between different objects. Sequence diagram highlights the message transfer behavior in chronological order. Communication Diagram highlights the organizational structure of the sending and receiving messages between objects.

Optimization

putting forward clearly the interaction between system and actor: Some roles' performing doesn't have to through our system, so remove them.

Merging postman and dispatcher: Because the function between such two roles is repeated, there is no need to distinguish them (such as checking the goods, etc.).

Making the functions of system modular: The system is divided into five main modules (delivery, receipt, accident, return & exchange, track), according to the five modules to analysis of the process.

Designing system architecture and optimizing the user interface.

Design Considerations

Business Assumptions

For customers, we offer a service that check order status, users can view their orders via the app on mobile phone, the status of orders will be updated by each postman. If the goods are lost, the customer can be no reason refund if goods are found damaged, the customer can choose to return or exchange goods. When the goods have arrived, the postman will make an appointment with customer, the customer pickup and confirm the delivery time, if the customer cannot pickup by appointment, they will arrange other arrangements or the customer let his(her) friends or families to take. We also provide choice of payment service, customers can choose to pay cash or credit card, and the payment completed, the delivery person will provide electronic signature tools like iPad to complete the electronic signature.

For postman we provide services to track goods, if goods are found damaged or lost, they will send text messages to the customer.

Business Constraints

Since there are many unexpected situations, so there will be a lot of constraints, when delivery staff find the goods miss or damaged, but he(she) unable to contact the customer leading to the customer can not get the notice, or when the goods arrive, delivery staff unable to contact the customer, he(she) can not schedule delivery time, delivery staff even have delivered goods to the designated location, the customer for some reason does not come to pick up, it will lead to the delay of delivery, if more than a few days the customer do not pick up, the goods will be automatically refunded. When sign the signature if the electronic equipment fails to respond will result in unable to sign, there may be a scene that customer has paid but can not pick up. So all our work is based on the conditions are satisfied.

System Environment

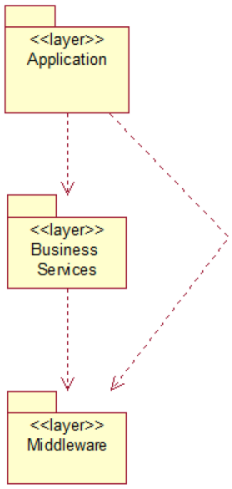
Software: iOS, android client, Web, Oracle database, Hadoop
Hardware: Server, Sensor

System Assumptions and constraints

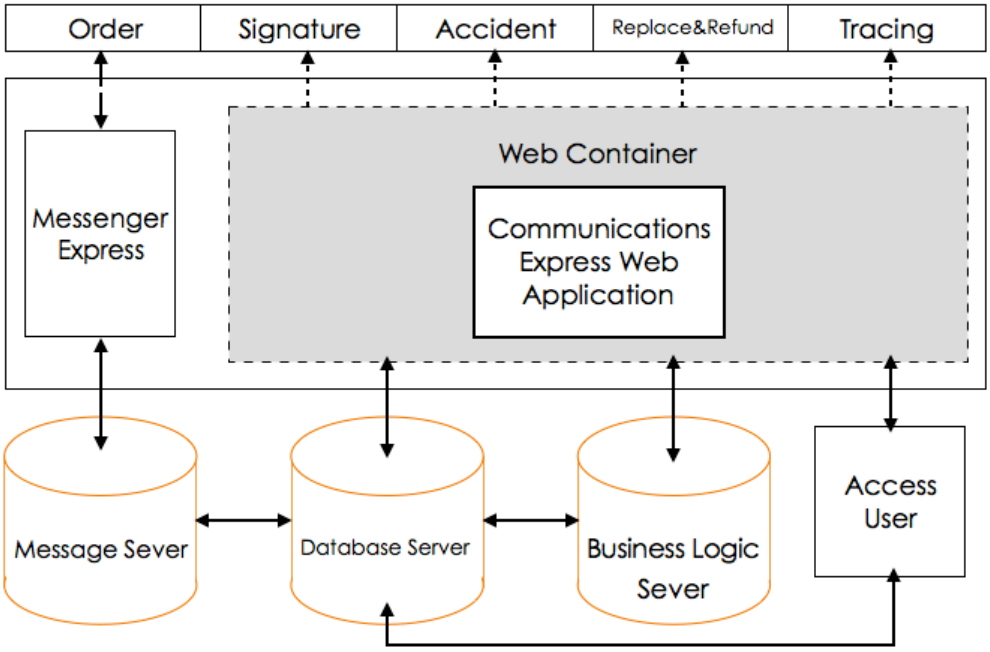
We are a B2C system, we achieve the client app, customers can check their order status on the app and with delivery staff to complete the reservation and other functions. Postman operated by web, postman tracking order status on the page. Postman modify shipment status information on the Web, etc.

Architecture

No matter how many projects throughout the company, we need to precipitate the most basic framework, which in general contains the core of the sub-library sub-table rule, unified database operations class library, unified communications, unified logging class, uniform encryption algorithms, uniform basis services SDK, public utility classes and more. Corporate Structure The framework is used to define the most basic design, the unified architecture-based technology and project specifications, interception and monitoring the most basic core recalls.



Because the system need to realize 5 function models，we choose using web to present our data。The system Architectural Graph is as follow:



We use web technique to realize interact between underlying server and front end, meanwhile we get feedback from front end to do the verify function. And the order information is transferred by message middleware to the server in the delivery system.

Sub-libraries sub-table

Sub-libraries sub-table is the most conventional splitting architecture solution. Usually splitting different perspective from a business perspective, such as the user's perspective and business perspective. Of course the premise is also needed to support business or other technical forces, not appear or across multiple sub-libraries solve split or divide table after table queries and result of the merger issue. Former sub-libraries sub-table also needs estimated by capacity and performance. Sub-libraries sub-table also often encounter the global id, or distributed id increment and the only problem, they must be fully considered advance in design and architecture level.

Distributed Cache

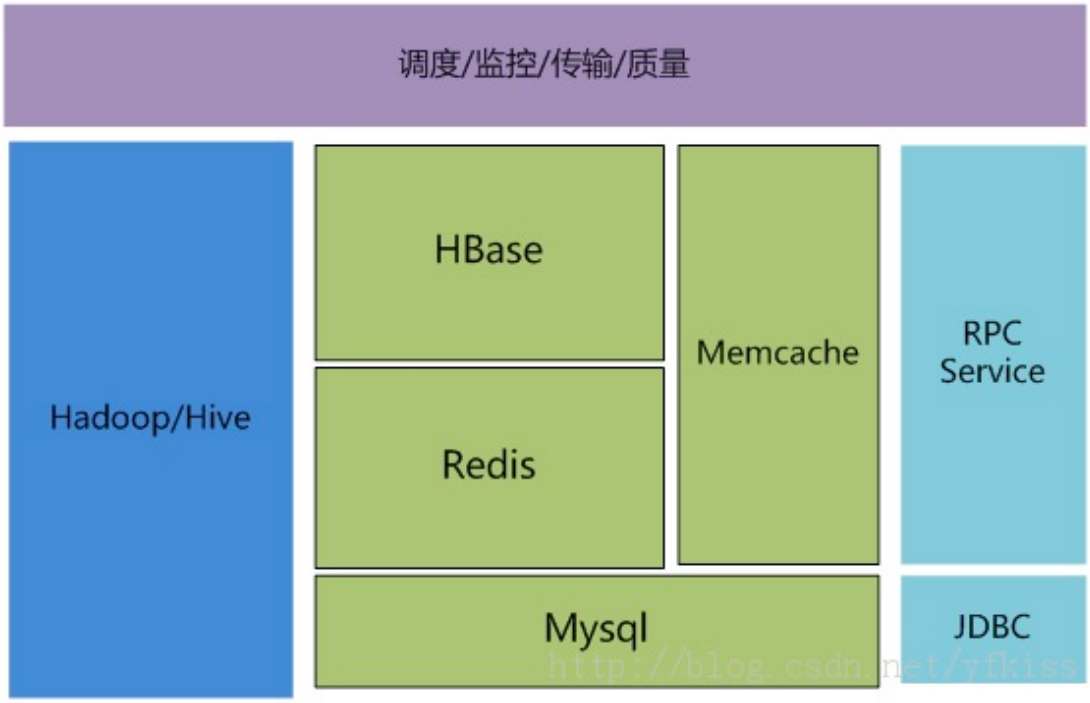
It is important to solve the core problem of a few pages of the cache, such as home pages and lists, so as to solve the high frequent frequent queries a database problem. In general contents of the cache, the more detailed the better, so the cached capacity will be large, the effect on database performance optimization naturally very good. However, the finer the cache, the more detailed work on cleaning up the cache, it is easy to forget that the process of writing code in clearing the cache, and the impact and user experience will be very bad.

This situation may be resolved in two ways. One is the architecture has reached the level of service and modular, each module handles only its own associated cache. Such as customer service, order services, delivery services, processing only relevant cache itself. Then the cache fine enough, of course, the code can also be more careful handling. Another is to modify the database or other levels of correction, batch clears the corresponding cache; because the particle size is very

thick, but may be a large number of available cache was cleared, causing part of the avalanche effect.

So we often think that the utility of the cache is easy, but with a good buffer required is based on business requirements and licensing the design cache structure, try to make good use of the cache, to achieve the desired performance; or we only use a small amount of coarse-grained cache, with defining cache expiration time, part of the code to clean up part of the way to the cache, which can ensure high performance hot page; however, in this case we still have to pay attention to the cache entry is not too much, coding specifications management.

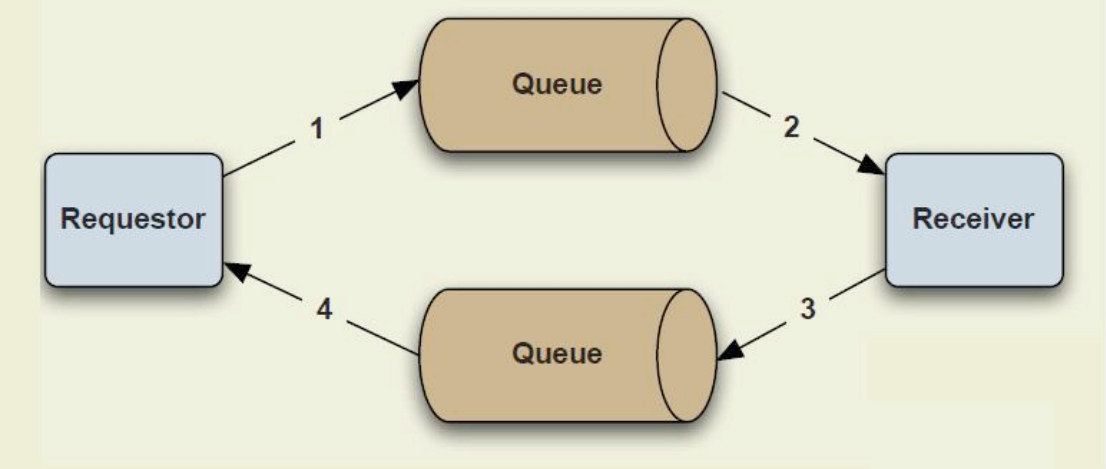
At the same time, we note that there will be some features of the cache operations, some cache with high hotspot feature, some cache with instant hot properties, some cache can be lost, and some try to ensure that the cache is not lost (risk of avalanche effect). So we want to depend on the actual business using different storage media. For example, redis, memcache, ssdb etc. scenarios are slightly different. And as the company level cache middleware should be appropriately shielded these storage features, the best you can do seamless configurations, along with load balancing, performance monitoring, and so on.



Business Message Queue

Mainly to solve the asynchronous processing high-reliability messaging services function. This message queue must have the following: carrying high concurrency, not allowing the loss of business news, business news must be able to support a large number of super-accumulation and stable support back messages. Usually the company might consider an enterprise service bus (ESB), but for a large number of e-commerce and instant messages piled high concurrency requirements, may not be appropriate, but a lot of things ESBs contains much belonging heavyweight solution, more suitable for general enterprise project such as the enterprise's internal management systems. Of course, some companies may also use activeMQ, RabbitMQ, etc., will

be selected according to specific scenarios and real business needs. For example, some memory messaging middleware does not support persistence, not support message accumulation, not support the message back, in fact not fitting the company's current business scenario, therefore give up or to use part of the scene.



Task Scheduling Platform

The main task is to solve business back-end mount, isolation, time scheduling, task error alarm. Associated son tasks can do the task automatically assigned tasks and coordination of resources, failover and balancing tasks. Then the Web crawler, report analysis, elastic computing and other resource-based task can be applied.

Unified monitoring platform

Each company will fetch and integrate some performance indicators, business logs, error logs, consuming performance, traffic and so on to the monitoring platform. For different business, performance parameters and indicators point concerns are different. So for long-term development, monitoring platform is a key role in order to the stability of the whole distributed system. Large distributed electronic commerce system itself is a very important part of monitoring platform.

Service Center

Mainly to resolve the problems such as synchronous call between projects, public API project sinking, load balancing remote call services, performance monitoring, early warning, and so on. Management Services is essentially a current, public, coordination, operation and maintenance, etc., to meet the architectural design business SOA. In particular, for the business split refinement, modularity, decoupling synchronization in the future, it plays a key role, similar to Taobao HSF.

Configuration Center

Mainly to resolve configuration aggregation and unified management of multiple projects, along with the configuration update. The company projects that have multiple projects, multiple projects also have multiple services, multiple background tasks. these programs are independent

from others, and some of them need load balancing, but they often require using the same set of configuration system. If each program is configured with a configuration information, then migrate a configuration information or updates in the future need a great difficulty for operating. Therefore, configuration management gathered in large projects is necessary. But it can also be achieved hot swap, automatic failover, load and other soft distributed services management capabilities based on the configuration center.

Application Level Design

Class Diagram

Entity

It contains user, postman, customer, package, order and info six classes.

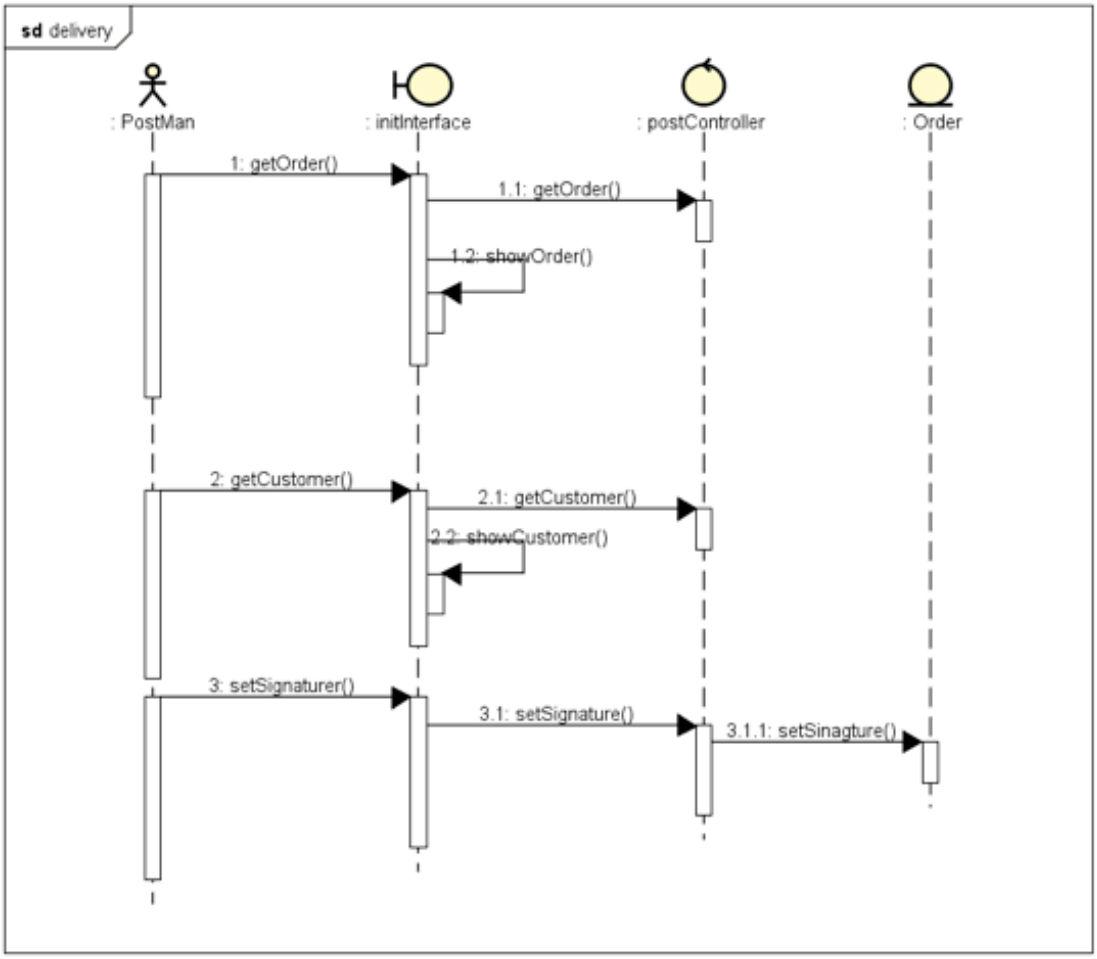
User is the basic class of the all user with the basic elements and functions. We also create two son class(postman, customer) which extend the User but add some extra elements and function basic on the role of it.

Package contains the basic information of the package, if it has one order we can create an order with the package message and some information else. By the way, we made a enum called state to draw a division between the different states including some other states.

Info is the class to draw a division between the different kinds of order,to handle it .

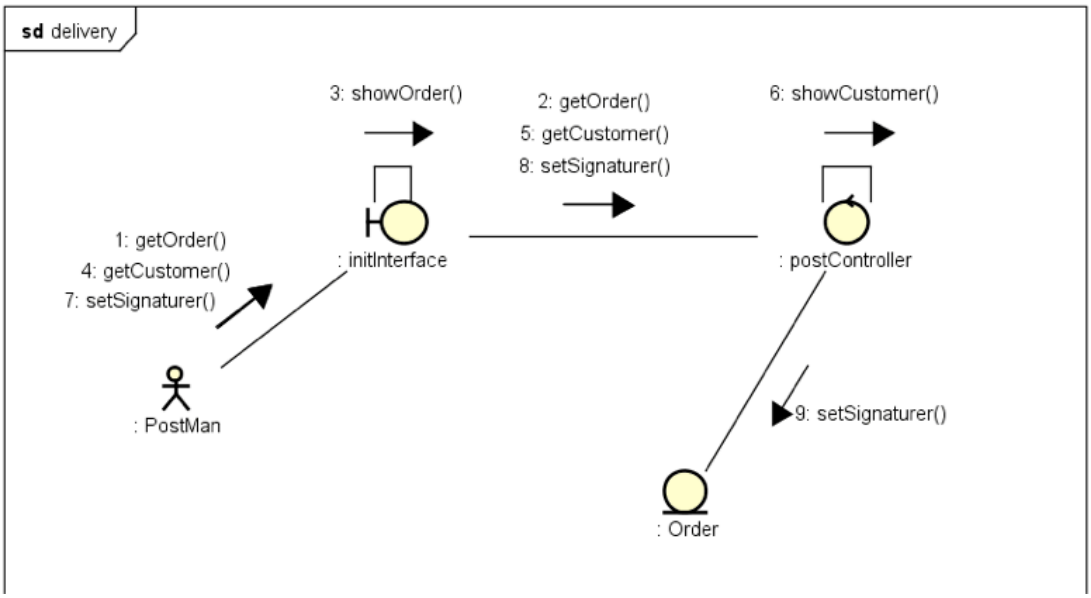
Delivery Module

DeliveryController is created to control the goods transportation process. postman can use it to get the information of the order, customer information and the time.



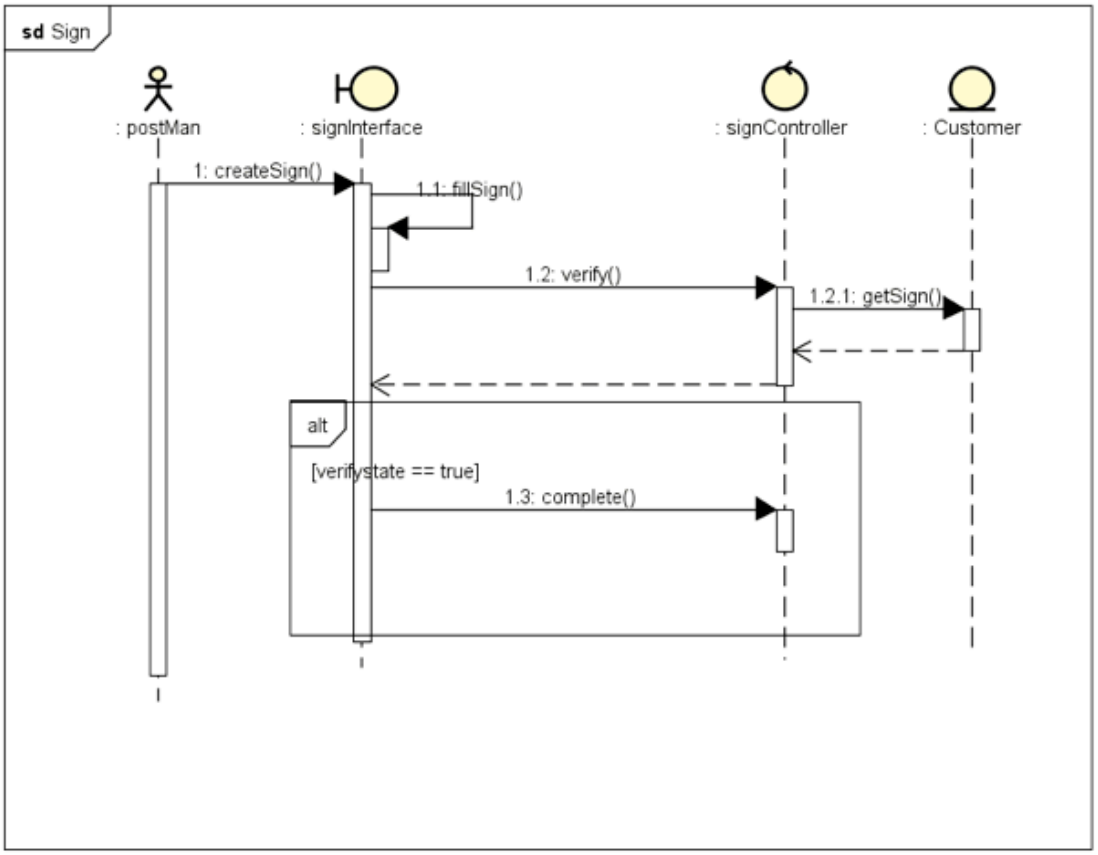
In this use case inside delivery, the postman can view the order in order to know where the package to be sent , to communicate with customers with some relevant information.After the

package was sent . the postman can get the user's contacts to determine the place and way to sign.After the user information is determined, the postman need to put the signature into the system to improve the information ,which is useful in the future verify.

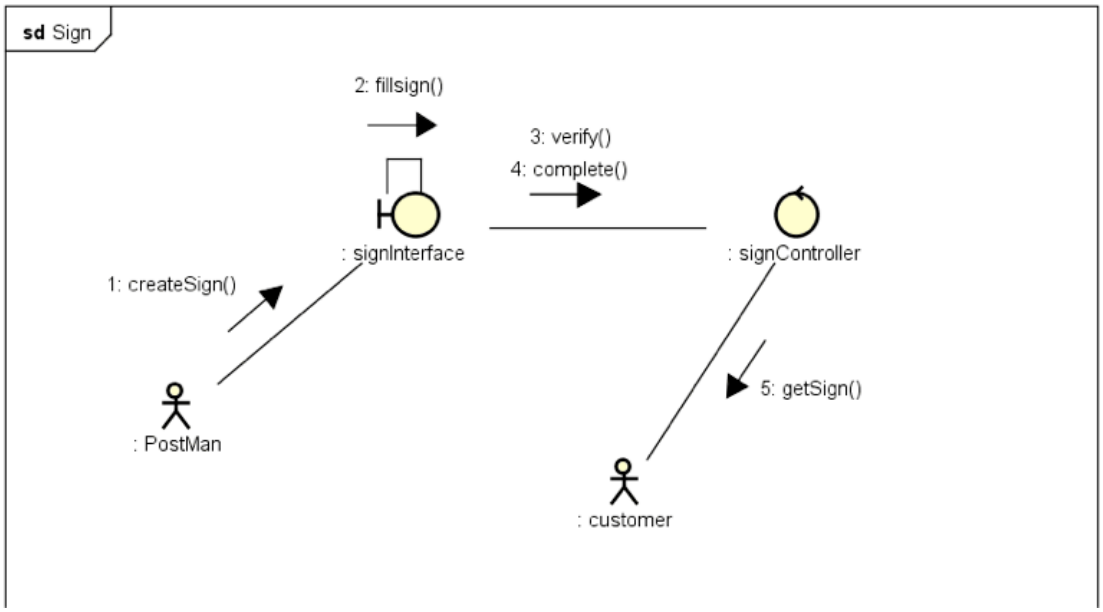


Sign Module

SignController is the class the control of the goods receipt process, customer can input fingerprint information or electronic signature and information can be checked by this class .



The postman create a user interface at the time of signing the delivery package and can choose to sign or fingerprint input manually. At the same time the system will verify the information with user information existed.If the confirmation, complete the package delivery process.

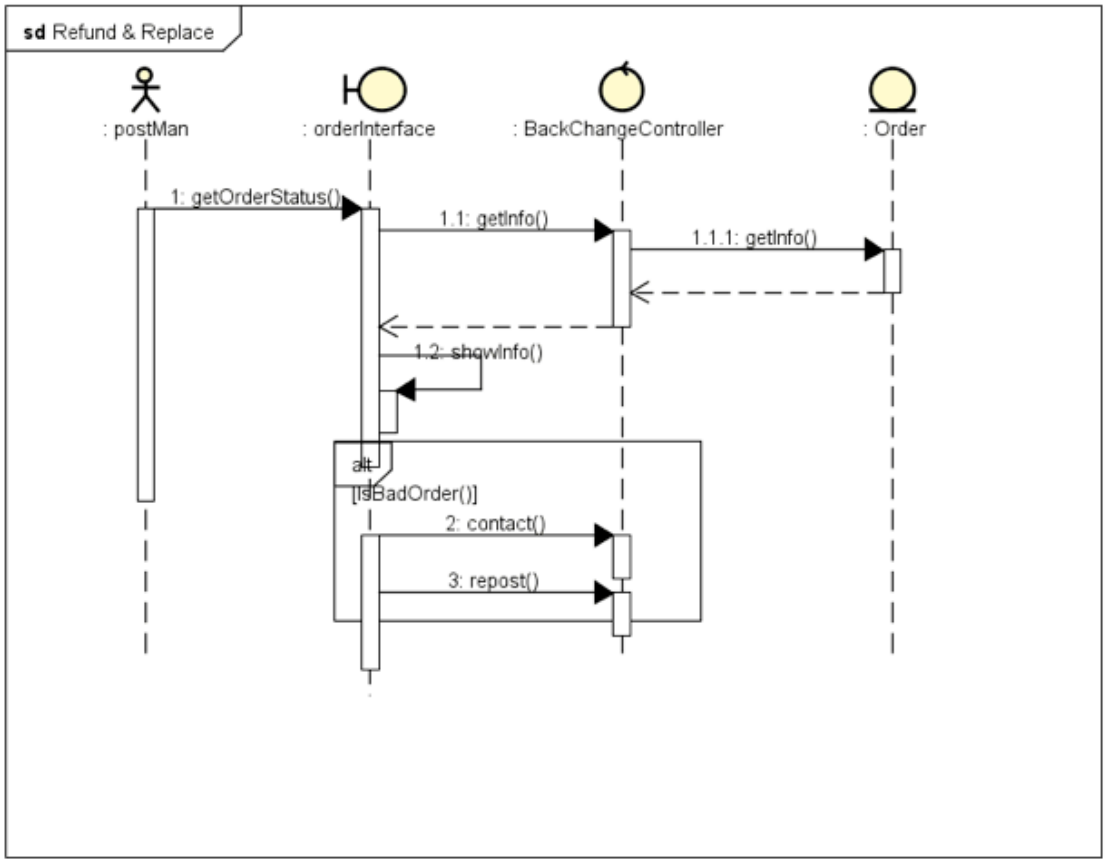


BackChange Module

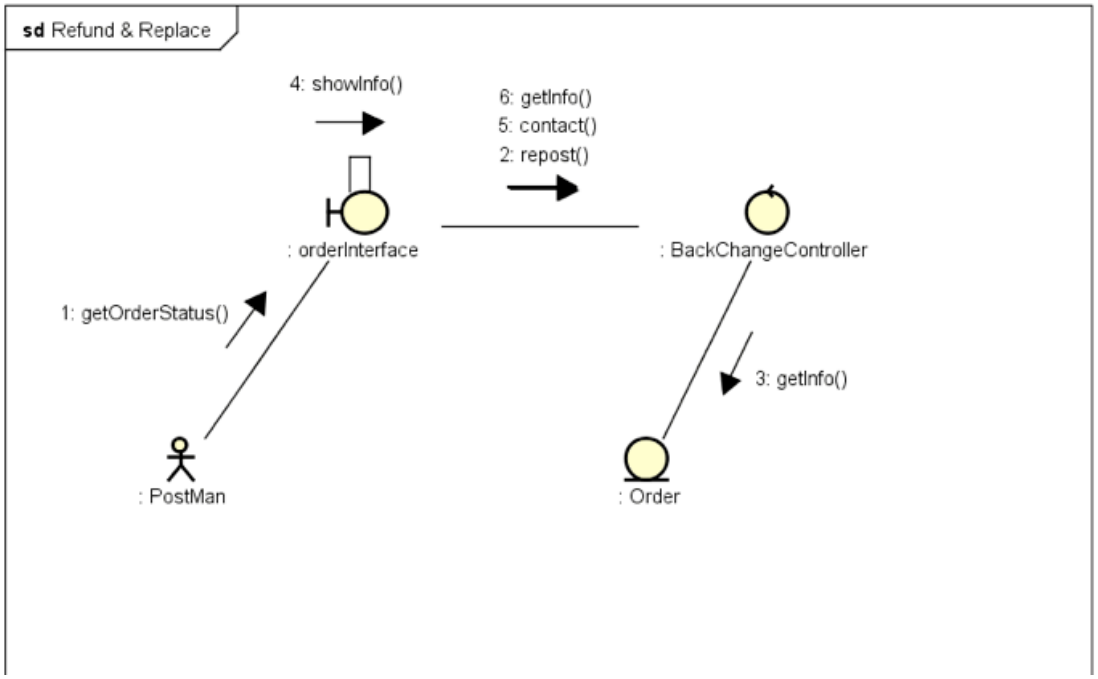
BackChangeController handles three abnormal condition.

-When postman in inspection, and find the goods lost or damaged, then he should interact with the system for replacement

- after receiving the goods, the user requests to change goods
- after receiving the goods, the user requests to send goods back.

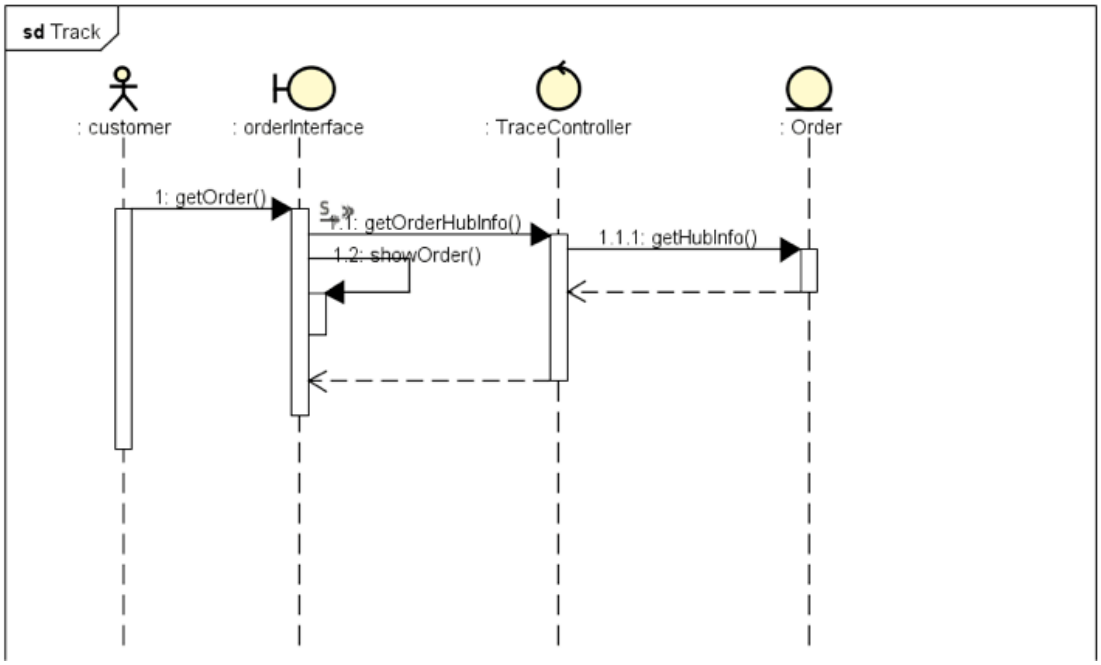


In the Refund and Replace use case ,Postman can check the status of the goods in the transport process,If found damaged or lost goods and other unforeseen circumstances, postman or customer can apply for re-shipment to the system.In this process, postman should contact with customers to explain the situation to reach.

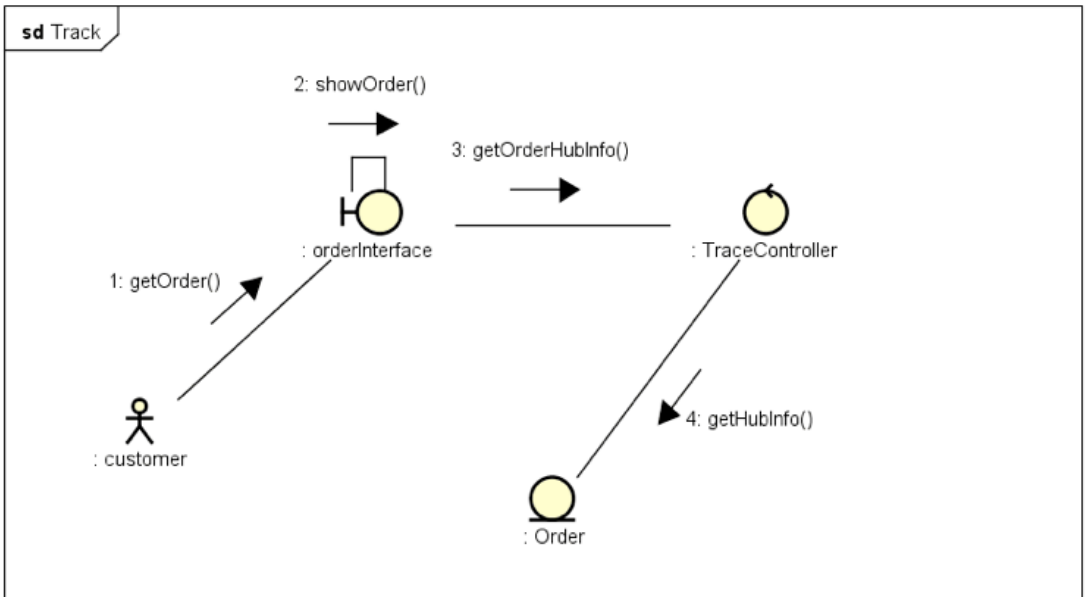


Trace Module

Tracecontroller establish the monitoring of the order status , and information to return



In the track trace use case.Users can view the status of the package to track the package is intact and the positions it has reached and to be reached.



Interface

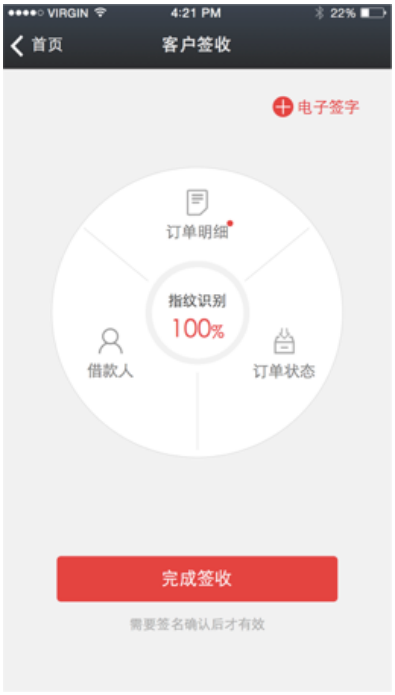
Each interface corresponds a controller. they are the bridge between user and controller for the information transmission and feedback.

User Interface Design

Delivery Screen

we can search for orders in this page on the right, there are some simple information of the order and you can click it to details page. and on second page we can see all the orders divided by the status finished or unfinished.

Sign Screen



customers and postman can find the tracking information form the first page, they will know where is package and who has it.

BackChange Screen

when customers are unsatisfied with the package they can return it by selecting the reason and writing some further information in the note. and click the button. system will handle this application.

And when the package is damaged or lost during the delivering, postman can submit an application to system.

Trace Screen

and when customers receive their package they can sign their digital signatures on the other page and if necessary they can also input their fingerprints to the system. after all these processors they can complete this model.

References

Kevin C.Dittman System Analysis and Design Methods, Seventh Edition [M]. the USA， Lonnie D. Bentley ,2008.

System Analysis and Design Methods introduces the method of system analysis and design, we mainly refer to the chapter 10 and the content of chapter 18 for object-oriented system analysis and design. The book also provides a current available on the market, the rich example of system analysis and design, this is design reference for us.

Craig Larman Applying UML and Patterns, Third Edition[M]. China Machine Press,2006

Applying UML and Patterns provides a feasible method of quick start, we mainly refer to the content of chapters 9 and 10 for object-oriented system analysis and design, the two sections for us to identify attributes of a class, the operation, the relationship between the class and help us a lot to draw sequence diagrams.

《电子商务系统规划与设计》徐天宇

The book details how to plan and design e-commerce system, introduced planning e-commerce systems, focusing on strategic planning and corporate e-commerce, e-business systems planning, planning and e-commerce sites, e-commerce payment system planning and e-commerce system security planning design e-commerce system needs to complete the overall system design and system design platform, Also details the structured approach of e-commerce systems analysis and design.

《基于 UML 的电子商务系统建模研究与应用》夏克付

The article includes software tools for UML application in the field of analysis and research on the characteristics and architecture of e-commerce system were analyzed. At the same characteristics and architecture for e-commerce system, analyzes the necessity of e-commerce system modeling, analysis of the key technologies of e-commerce system based on UML modeling, combined with UML modeling process proposed UML-based e-commerce system.

Members and contributions

Bo Qin *user interface Document class diagram*

Jiaqi Zhang *class digram*

Juyi Zhang *ppt*

Jingru Tan *communication digram*

Huilin Kang *sequence digram*

Rihui Huang *document*