ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 659-663
doi: 10.18178/wcse.2017.06.115

An After-sales Service Management System Based on Activiti

Meiyu Chen [1] [+]and Xiangdong You [2]

[1] School of Information and Telecommunication Engineering

[2] Beijing University of Posts and Telecommunications, China

Abstract. In order to raise the efficiency of after-sales service and solve the problem of irrational allocation of resources, we developed a web site to assist workers to offer after-sales service, and this paper mainly describes the solution to achieve this goal. Users can achieve those services provided by the platform by browser or through an APP on smart phone, both are supported. Activiti is used as the basic framework because it can provide us a series of service components that can be used to develop the configurable process definition module and form design module. Spring, Hibernate, Spring MVC and other framework are integrated into the project to make the development much easier. Further more, the application of the Three-tier architecture, Command Mode and Component-based principle has also made the whole project Loosely-coupled and tidy, which ensured the maintainability and stability of the system.

Keywords: after-sales service management system, activiti, MVC, component-based principle, command mode, framework

1. Introduction

In order to maintain their reputation, the vast majority of merchants offer customer relevant after-sales services, including consultation service, fault diagnosis service, maintenance service, and so on. Traditionally, customers get access to those services following this way: firstly, they should ask dealers for help; secondly, the dealer will report it to the merchant which has produced the product; then manager of after-sale service department will assign related technicians to handle this problem. In some cases, technicians should apply components from relevant department to fix the equipment. This procedure involves communication and collaboration among multiple sectors, however, inter-sector communication will generally take a long time, this makes the entire service period extended, not only reduces the quality of after-sales service, but also raise the cost of after-sales service.

Additionally, handy query for information of the product should be offered to relevant person, because those information may help them to make decision. For example, maintain records may assist the technicians to find out causes to the failure of the equipment and fix it in less time. However, traditional management of those data can't offer this kind of query, because most data are recorded by paper work order or independent spreadsheet. In the process of demand analysis, we learned that the utilization rate of those data is actually very low, which made a great waste.

To solve those problems listed above, we developed an after-sales service management system based on computer software technologies. Specifically, we used work-flow to simulate the whole procedure of after-sales services, work-flow management promise a new solution to an age-old problem: controlling, monitoring, optimizing and supporting business processes [1]. What’s more, grid is used to build this platform. Grid is a new paradigm of Internet computing to share distributed resources and collaborate among them, a web service-based approach for Grid can improve the extensibility and interoperability of Grid system [2].

[+] Corresponding author. Tel.: +86 13701112344; fax: +86 01622821711.
E-mail address: 13701112344@163.com.

Different roles from different departments were united together by our platform, users can achieve the same ProcessInstance even when they present at different places. Many platform like this already exists, but most of them are not flexible enough, and few of them support data access. Out of this point, we developed our platform, and it has some features that shouldn't be neglected:

- Firstly, unlike fixed work-flow provided by most platform, our platform support flexible process and form modification, administrator of this platform can modify the form and the work-flow according to needs. What's more, every process deployed on this platform is configurable, administrator can bind form, events, message or other things to each Node through configuration module.
- Secondly, our platform allows rollback, withdraw and loop assignment operation, which optimized the user experience.
- Thirdly, users of this platform can delegate their colleague to complete tasks assigned to them if needed.
- Most importantly, this platform has also integrated data management module. Users can request for kinds of information according to needs, for example, technicians can achieve records of the faulty equipment, and those records may help them to fix the problem.

2. The Use Case Diagram of the System

In order to meet the requirements, we had delivered a series of survey, and carried on thoroughly analysis according to the work-flow of the after-sales services. Finally, we divided the whole platform into seven module according to the general requirements analysis, which includes information portal, account management, task management, forms management, progress management, user group management and personal affairs management module. As the work shifts from design to implementation to testing and deployment as the phases progress, those module referred above was subdivided into smaller modules as shown in the following diagram(“Fig. 1: The Use Case Diagram of the System”). The process of further division to fine-grained modules can not only get the developer understand the true demands of user, but also help them to develop a product that covers a complete range of requirements, and enhance the acceptance chances.

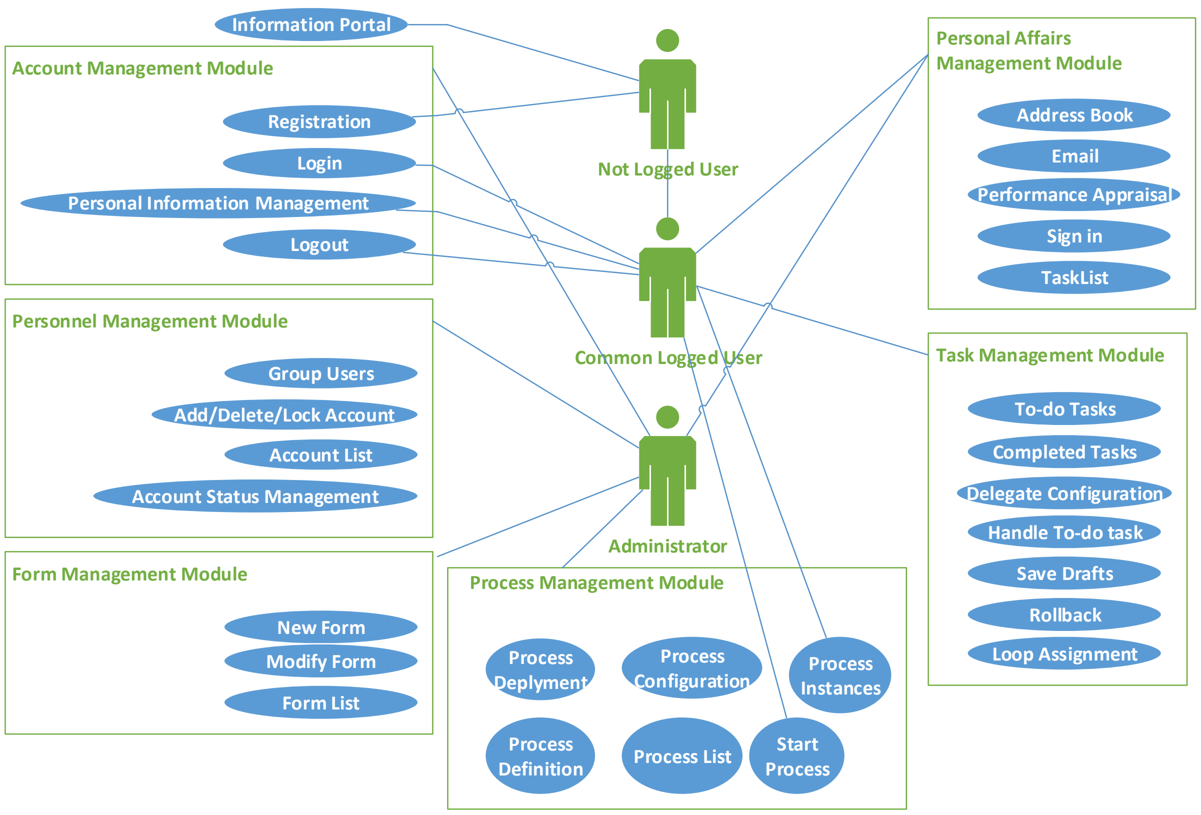

Fig. 1: The use case diagram of the system.

3. The Solution Architecture

Considering features of the system, we choose following technologies to develop this system.

1) Development and Deployment Environment

- Development Environment: JDK7+MyEclipse+Maven3. Apache Maven was mainly used for managing jar.
- Deployment Environment: Tomcat7+MySQL+JVM. Tomcat is an open source server, MySQL is a relational database.

2) Work-flow Engines: Activiti. Activiti is battle-tested Business Process Management. It supports open standards with BPMN and DMN. Developer can use it's API such as ManagementService, HistoryService, RuntimeService or other components to implement their own work-flow. Application of those components provided by Activiti can simplify and speed up the development process. The Tripartite Framework: Spring+Spring Web model-view-controller(MVC)+Hibernate. The Spring Web MVC framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, locale, time zone and theme resolution as well as support for uploading files [3]. It can be used to develop Restful Web sites and applications. The Spring framework supports Dependency Injection and Inversion of Control, developers can use it to manage beans. Developers can also use the spring-aop module to incorporate behavioral information into their code, this module supports source-level meta data functionality. At present, database middle-ware is one of important technology in integration of data in heterogeneous databases [4]. As an Object/Relational Mapping (ORM) framework, Hibernate is concerned with data persistence as it applies to relational databases (via JDBC) [5]. Except for those framework listed above, JSP and JavaScript was also integrated into our project, they are used for building templates for View layer. Model was rendered by selected template before they are passed to user.

3) Other Technologies: Flyway+JSA+DBCP. It’s difficult to deal with heterogeneous database replication with little overhead in large applications [6]. Flyway is a open source database migration framework for Java, it brings structure and confidence to the evolution of your database schema [7]. JSA is used for program compression. Many Apache projects support interaction with a relational database, creating a new connection for each user can be time consuming, DBCP provides a "pool" of open connections, and connections in the pool was shared between all of the application's current users, which solved the problem successfully [8].

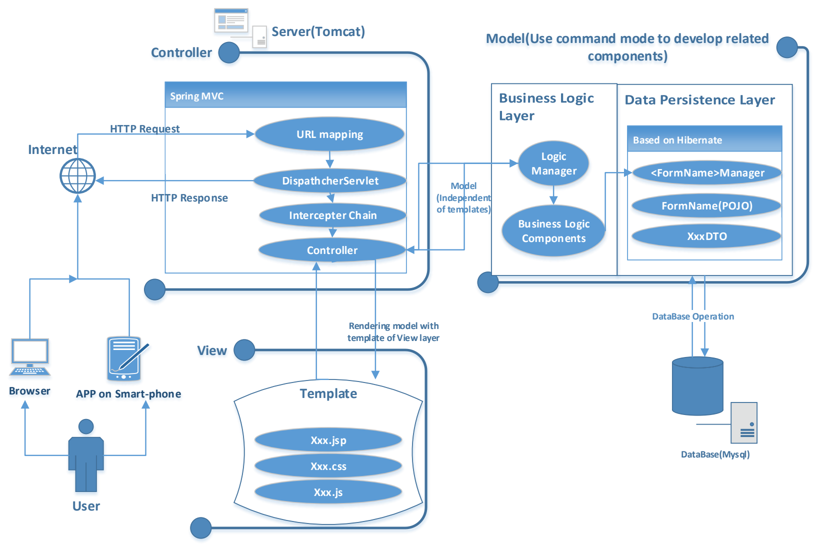

Fig. 2: The system architecture.

As is showed in Fig. 2: The System Architecture. Users can get access to services provided by this system by browser or smart phone, both are supported. The HTTP request made by users will be intercepted by the DispatcherServlet of Spring MVC, and then an appropriate handler is searched for handling this request, and the execution chain associated with the handler is executed in order to prepare a model or rendering the model. In Command Mode, handler acts like a invoker, it accepts request and call for command to handle the problem, then the receiver called by the command will conduct real operations to get the requested data. In this system, business logic components act as concrete command, and it may calls component of data lasting layer to get data from the database. At last, model obtained after a series of operations listed above will be rendered by templates of View layer, and users will receive it as HTTP response.

4. The Work-Flow of After-sales Service

As shown in the following figure(Fig. 3: The Work-flow of After-sales Service), users of this platform can start a process instance applying for after sales service provided by the manufacturers, and then the dispatcher may assign proper maintenance technician to fix the problem. If the technician accept the assignment, he must fix the problem in the specified time period and fill the form bound with each phase he is responsible for. In some case, technician can also refuse the task assigned by the dispatcher, for example, the technician can refuse the assignment if he is on holiday or he has other tasks that's much more urgent. The dispatcher can modify the assignment if needed, if so, the old execution flow will be stopped and a new execution flow will be created, what's more, historic data will be saved in the database. The record will be submitted to the performance appraisal department after the maintain technician has completed the task, then somebody of this department can pay a return visit to the customer about the maintaining process, and they can make a record about the return visit. The support for performance appraisal can motivate their employee to work harder. The process instance come to an end after the performance appraisal, and this is the work-flow of after-sales service supported by our platform. Administrator can modify it according to needs, the modification is very convenient.

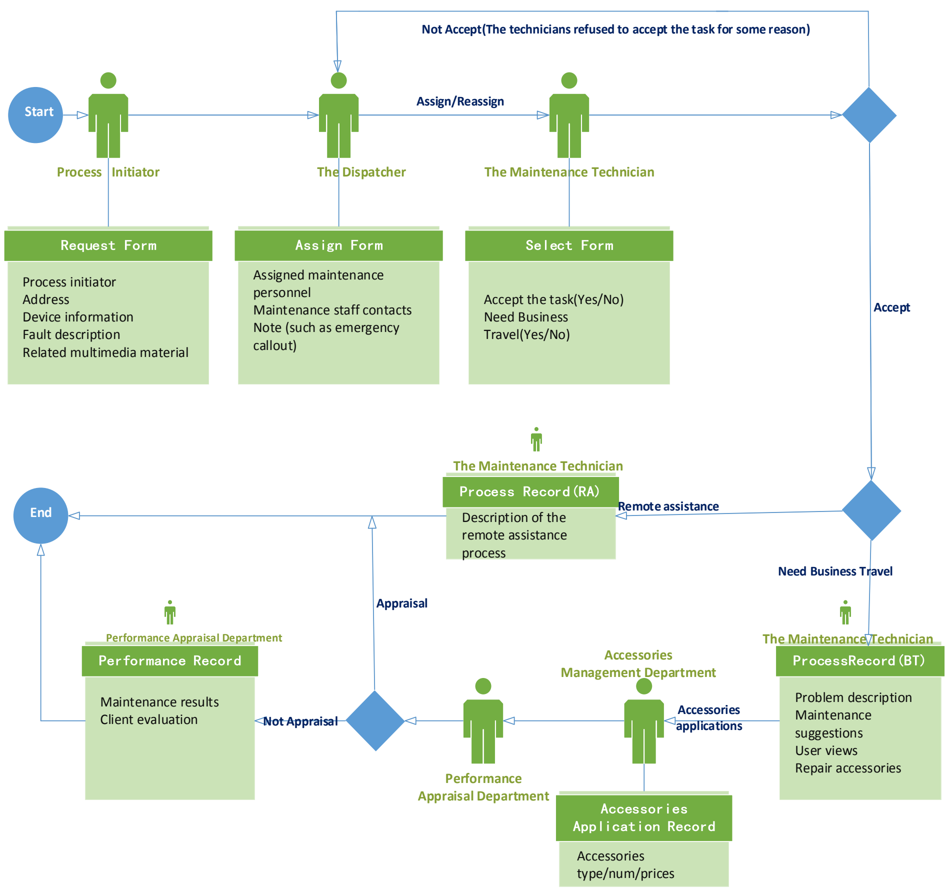
Not Accept(The technicians refused to accept the task for some reason)
Start
Assign/Reassign
Process Initiator
The Dispatcher
The Maintenance Technician
Request Form
Process initiator
Address
Device information
Fault description
Related multimedia material
Assign Form
Assigned maintenance personnel
Maintenance staff contacts
Note (such as emergency callout)
Select Form
Accept the task(Yes/No)
Need Business Travel(Yes/No)
Accept
The Maintenance Technician
Process Record (RA)
Description of the remote assistance process
Remote assistance
End
Need Business Travel
Appraisal
Performance Appraisal Department
Performance Record
Maintenance results
Client evaluation
Not Appraisal
Performance Appraisal Department
Accessories Management Department
Accessories applications
The Maintenance Technician
ProcessRecord (BT)
Problem description
Maintenance suggestions
User views
Repair accessories
Accessories Application Record
Accessories type/num/prices

Fig. 3: The workflow of after-sales service.

5. Future Work

The platform provide users an efficient way to manage after-sales services, it has been put into usage now. However, in order to make it more popular, we will develop more modules based on this version. What's more, safety of the those data managed by this platform should be enhanced in the future.

6. Conclusion

This paper introduces to reader an after-sales service management system. It consists of several parts: the first part introduces the background of this project and listed several advantages of this platform compared with other similar platform, the second part describes the user case diagram of this platform, the third part of this paper lists the technologies used to build this platform, and section four shows reader the work-flow of after sales services. Hope this paper can give readers some enlightenment, please don't be shy to contact us if you have any questions.

7. Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor, XiangDong You, for his instructive advice and useful suggestions on my thesis. I am also indebted to all my friends for their direct and indirect help to me, special thanks should go to them for their comments on the project and the thesis.

8. References

[1] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management[J]. Systems and Computers, 1998

[2] HU Chun-Ming+, HUAI Jin-Peng, SUN Hai-Long. Web Service-Based Grid Architecture and Its Supporting Environment[J]. Journal of Software, 2004-07

[3] Rod Johnson, Juergen Hoeller, etc. Spring Framework Reference Documentation[EB/OL]. http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/. 2016

[4] Xiaohui Yao, Hui Wang, Yajun Zhang. The Application of Hibernate in System of Heterogeneous Databases[J]. Computer Knowledge and Technology, 2011, 07(8)

[5] Red Hat, Inc. Hibernate JavaDoc (5. 2. 9. Final)[EB/OL]. http://docs.jboss.org/hibernate/orm/5.2/javadocs/. 2016

[6] Xiaohang Li, Xiaopeng Hu, Gang Li. Improvement and Implementation of Heterogeneous Database Replication Approach[J]. Computer Engineering, 2009, 35(8)

[7] Axel, Fontaine. Flyway - Database Java Migration Open Source Framework[J]. Method & Tools, 2012, **20**(3): 44-50

[8] The Apache Software Foundation. The DBCP Component[EB/OL]. http://commons.apache.org/proper/commons-dbcp/. 2017