A Unified Modeling Language-Based Design and Application for a Library Management Information System

Jianhu Zheng[1], Yunqing Feng[2], Yun Zhao[2]

[1] *Transportation Engineering Institute of Minjiang University, Fuzhou 350121, China*
[2] *School of Economics and Management, Beijing Jiaotong University, Beijing 100044, China*
Email: zjianhu1028@163.com   yq.fung@gmail.com   zhaoyun0614@126.com

Abstract: This paper firstly introduces the main content of the Unified Modeling Language (*UML*) *and proves that it can transmit information among the users, the developers, the designers and the managers efficiently, which improves their collaboration capabilities and greatly increases the degree of industrialization in software development projects. Secondly, a library management system development and design is carried out, based on UML modeling mechanism to analyze a simple library management system. Thirdly, a demand analysis mode of the management system is built with the help of a case diagram and an analysis diagram after analysis of a simple library management system, using UML modeling mechanism. Then a book lending management subsystem has been designed in the library management system by a design class diagram and a sequence diagram. The design process indicates that as a modeling language of software engineering, UML has a very good application prospect.*

Keywords: Object-oriented systems analysis and design; UML; library management system.

1. Introduction

With the rapid development of computer technology and the continuous improvement of the information level, the production scale of software is increasing and the traditional software modeling language has been unable to meet the demands of software development. Therefore, how to develop high-quality software within a reasonable period of time has become an urgent problem to solve. At the same time, people have also higher requirements to the design, maintenance and management of the software development, which includes shortening the software development cycle, improving the software quality and safeguard of the sustainable development of the software. All these problems can be solved gradually with the introduction of the Unified Modeling Language.

The Unified Modeling Language (UML) [1] is a standard, object-oriented, component-based software system modeling tool. It is a tool used to draw a visual description for the software system model. In the past, dozens of object-oriented modeling languages are independent. The emergence of UML has solved the biggest problem of the software exchange. The use of UML enables various staff, such as the users, analysts, developers and designers to communicate efficiently and enhance mutual understanding, which can improve the efficiency of the software development and the quality of the software design. The scope of UML is not limited to the support for object-oriented analysis and design, but it o supports the whole process of software development which starts from requirements analysis. It helps the lopers to draw a clear model which is conducive to communication and makes the sftware development easy to implement.

2. Overview of the unified modeling language

2.1. Characteristics of the unified modeling language

UML is a standard modeling language for object-oriented technology, which evolved from Booch method, OOSE method and OMT method [2]. It is a graphic language to visualize, describe, build and document the software-intensive systems. It combines the popular object-oriented development methodology concepts and technology, and forms a standardized object-oriented UML. UML unifies the basic concepts, terminology and graphic symbols of the object-oriented modeling and describes the basic steps that must be followed in the modeling process. This modeling language provides a method to describe the concepts and graphics of the software system model from a different perspective.

The main characteristics of the UML are as follows [3]:

(1) UML collects the excellent ideas of Booch, OOSE, OMT and other object-oriented approaches. These ideas are the essence of the best object-oriented methods and rich practice of computer science experience.

(2) Although UML is currently the most advanced and practical standard modeling language, some of the notions still need to be fully verified in practice. UML is in development of continuous improvement as well.

(3) The process is an important part of a method. There is no concept for the process in the UML. So that UML is just a modeling language rather than a method. UML is independent on the process.. Users can use UML to model any suitable process. The selection of the process relates to different factors that affect the software development process, such as the type of the software development, the scale of the development organization. The users will be determined to choose the process based on the objective situation.

(4) Nevertheless UML modeling is still a relatively unified framework which contains the common elements in UML modeling process. Besides, it provides a great degree of freedom for the users in choosing suitable modeling techniques for their development projects.

(5) UML is independent on the process, but the application is still driven by the case and it uses the architecture as a center.

2.2. Modeling mechanism of UML

UML is a general standard modeling language which contains a static modeling mechanism and a dynamic modeling mechanism [4]. The static modeling mechanism defines the attributes and operations of the important objects in the system and the relationship between these objects. The dynamic modeling mechanism defines the time characteristics of the objects and the communication mechanism among them in order to complete the objectives and tasks. Using the UML for analysis and design of an object-oriented system usually has three steps [5].

(1) Requirements analysis phase

First the developers should communicate positively to analyze and refine the requirements of the system. List the involved operators in the system and identify all the use cases and roles to describe. And then to analyze the relationship between the roles and the cases and use UML modeling tool to draw the case diagram.

(2) System analysis and design phase

First identify all the needs of the system. Abstract the classes from the actual needs and describe the relationship between these classes. Establish a static model of the system to construct the system structure according to the demand. Create a dynamic model based on the used cases and the static model.

(3) Phase of system implementation

The model in the first two phases is actually created in a system logic aspect. This phase is to accomplish the physical realization of the system, such as executable files, libraries, tables, documents, etc. The model diagram needs to be built is a component diagram and configuration drawing.

2.3. Review of UML application

In the aspect of the characteristics and advantages of UML it has been widely used in various areas of the system design. D o r o n D r u s i n s k y et al. [6] have presented a new approach UML state chart-based assertions for developing libraries of temporal formal specifications. With the gradually increase of the book order catalog's usage, the library technician should have deeper understanding of the basic model and framework of the recommender and purchase system. L i T i e-c h e n g [7] has summed up the general idea of the UML modeling language and analyzed the requirements of the recommender and purchase system. Then he gives a specific model and framework of the system.

L v Q i-y i n g [8] has analyzed the library management system based on the B/S model. She uses the static and dynamic model diagram of UML and creates a model for the analysis results of the system. Z h a o Q i n g-y a n and Z h a n g C h a o [9] have proposed a practical requirement analysis and a modeling method based on the Unified Modeling Language. Z h o u G e [10] first briefly introduces the main concepts of the Object-based Unified Modeling Language, and explains the application of UML in management information system development. Z h u C h e n g [11] introduces the basic concept and modeling process of UML through analyzing the book management system and explains the practical applications of UML. G a o L i-p i n g [12] takes a library management system as an example and discusses the modeling procedure of UML in this system. Z h a n g L i-j i e et al [13] have used the design and modeling procedure of Akesu CCP School's LMIS as an example to discuss the characteristics of UML and UML's application in information system modeling procedure. Taking the books management system as an example, Y u Q i a n g [14] describes every stage of the system design based on UML and introduces the application of UML in the design of a databank system. C h e n S h i-c h u a n [15] put forward the object-oriented analysis and design method through the further research of UML. Combining the developed instance of a book information management system, they have carried on a research of the technical application of UML and explained the relevant knowledge of UML. This research has solved the difficult problem of exchange among the software designer and customer, accelerated the development process, and improved the quality of the system. C h e n M i n g [16] discusses the Rational Unified Process driven by use case, and gives a solution for capturing the requirement of a library management information system. Taking the development of a library management system as the background, C h e n g C h a n g-p i n and C h e n Q i a n g [17] completes a system based on UML modeling and achieves it with VB.

3. Library management system design and development process based on UML

3.1. System Design Overview of the UML

From the UML modeling mechanism we know that UML-based object-oriented systems analysis and design usually have the steps of requirement analysis, system

132

design and system implementation. The object-oriented requirements analysis is the exact model to obtain the user's requirements and establish the problem domain by extracting and sorting. But the design process is to change the requirement, which is obtained in the analysis phase into a low-cost, high-quality, abstract system implementation. It is a step-by-step expansion model from object-oriented analysis to object-oriented design. In other words, the object-oriented design is the process of using the object-oriented view to model.

The purpose of the system design has the following four aspects [18].

(1) To deeply understand the non-functional requirements which are connected to the programming language, building reuse, operating systems, distributed and concurrent technology, database technology, user interface technology and transaction management technology.

(2) To create the appropriate input and starting point for the subsequent realization of activities by capturing the subsystem interfaces and classes.

(3) To divide the practical work into different parts which are easy to be managed and try to develop the system by different development groups in system design.

(4) To establish the seamless abstract of the system and take the achievement as a direct refinement of the design.

3.2. Requirement analysis of the book management system

The requirement analysis is generally used to define the system function and the user interface. It makes the users understand the system function and the developers to know the system needs. The main job of the requirements analysis is to find the system use cases and set up the system requirements model. Its main outcome is the use case diagram and system participants. The library information management system is an electronic archives management system that uses the computer to process a large number of book information. This system must meet the needs of three types of users, which are the readers, the librarian and the system administrator. The reader and the librarian may include many individuals. The system administrator only includes one person. The readers' behavior is to query the personal information, to query the book information, to book, borrow and return the books. The librarians are the main users of the system. They take the responsibility of daily management and service operations, which include books ordering, new book verification, bibliography input, book registration, borrow and return registration, reserve registration, etc. The system administrator manages all the issues related to the readers and the books, which include the reader information management, the library management and the maintenance of the system. Among them, the system maintenance is the key point, which includes the maintenance of the user right and log, to add and delete users, background data management of the system, etc. [19]. The functional analysis diagram of the library management system is shown in Fig. 1.
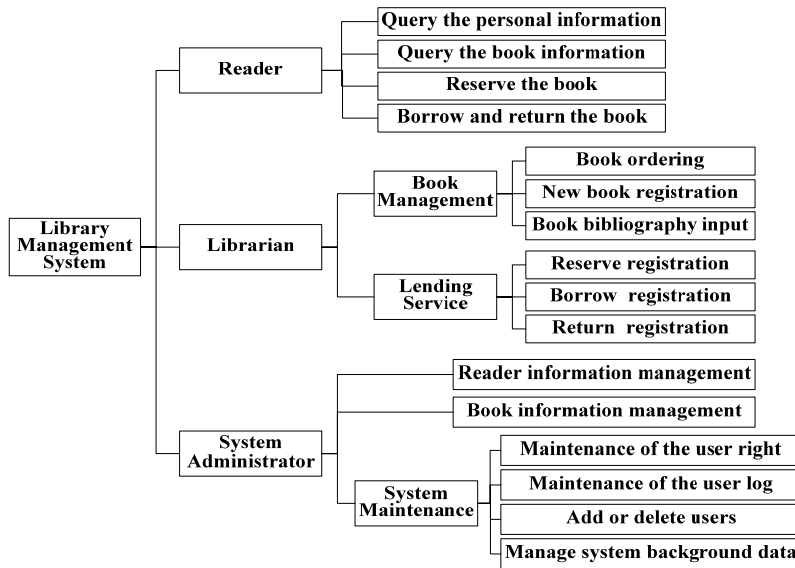
Fig. 1. Functional analysis diagram of the library management system

3.3. Model analysis of the library management system

3.3.1. Static model analysis

The static model describes the system from a static point of view according to the system structure. It defines the object and the class in the system and the relationship and internal structure between these classes, which are named attribute and operation class. In the object-oriented analysis, only the objects that are related to the problem description domain and the system function are generally considered. The objects of the class can be abstract in two aspects, book management and book circulation. These two parts can be divided into different classes, which are a book class (Lib_Book), a category class (Lib_ Category), an order class (Lib_Order), a form class (Lib_Form), a reader classe (Lib_Reader), a circulation book class (Lib_Cirbook), a department class (Lib_Department), a publish class (Lib_Publish), etc. The associated relationship of these classes is shown in Fig. 2.
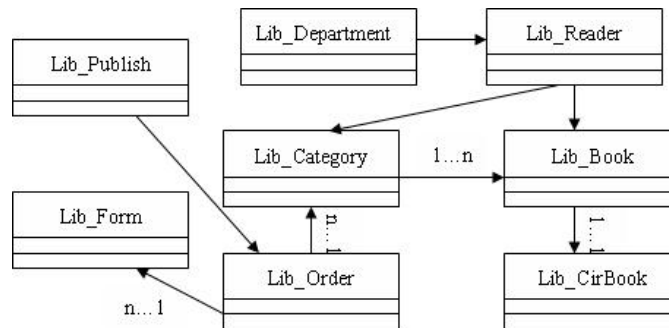
Fig. 2. The associated relationship of the classes in the static model

3.3.2. Dynamic model analysis

After the establishment of the static model, the construction of the dynamic model begins. The dynamic model is usually used to describe how the system function is realized. The interaction between different objects is described by a sequence diagram, an activity diagram and a collaboration diagram from different points of view [20]. The dynamic process of the book borrowing module in the system is shown in Fig. 3.
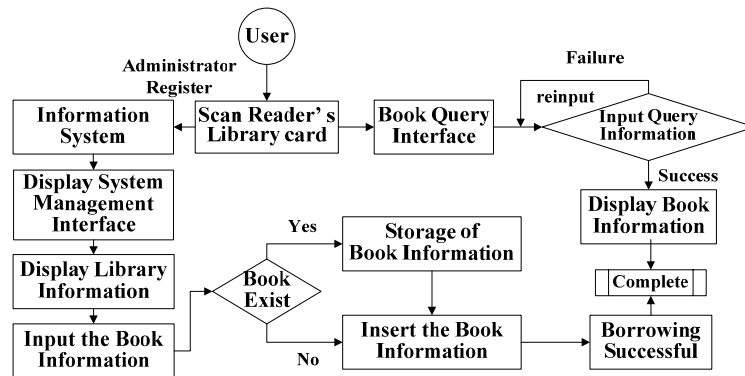

Fig. 3. Dynamic process of the book borrowing module in the system

3.4. Use case analysis of the library management system

Through the requirement analysis, we can see that this system has four activities, which are Lib_People、Lib_Librarian、Lib_Reader and Lib_Database. The Lib_People stands for the people that interact with the system. The Lib_Librarian stands for the librarian who takes charge of adding or modifying the book information. Lib_Reader stands for the reader who can borrow, renew and return a book. Lib_Database is the database of all information in the system, which can complete all kinds of system operation. The use case diagram of the library management in the system is shown in Fig. 4.

From the use case diagram we can see that the functions of the book management system are as follows:

(a) readers can query book information, borrowing information and reserving the book;

(b) readers can borrow, renew and return the books through this system;

(c) the librarian can handle the problem of book reservation, book lending, book renew and book restitution;

(d) the librarian can also use this system to provide the readers with overdue books.

The library management can be divided into book lend management, book return management book reservation management subsystem. The analysis chart of the book lending management is shown in Fig. 5.
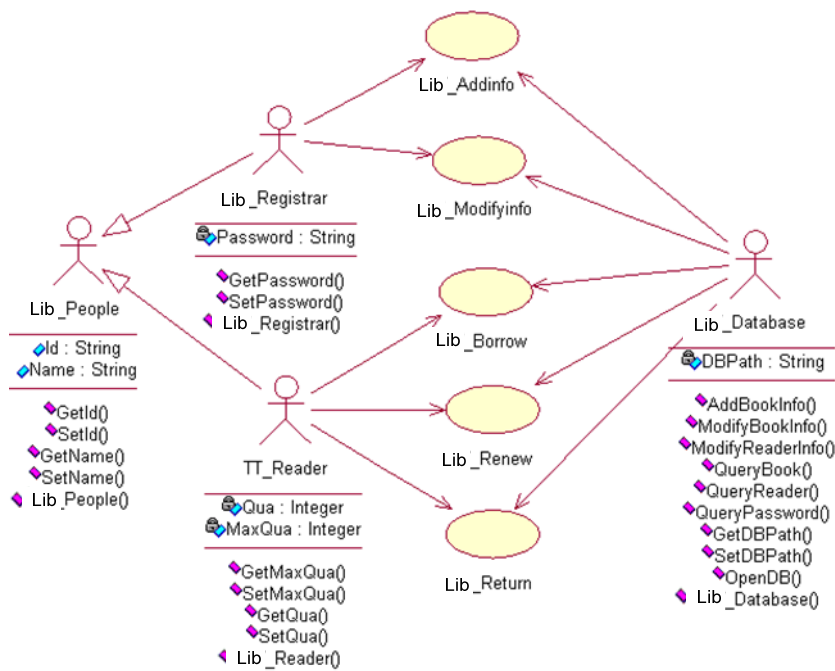
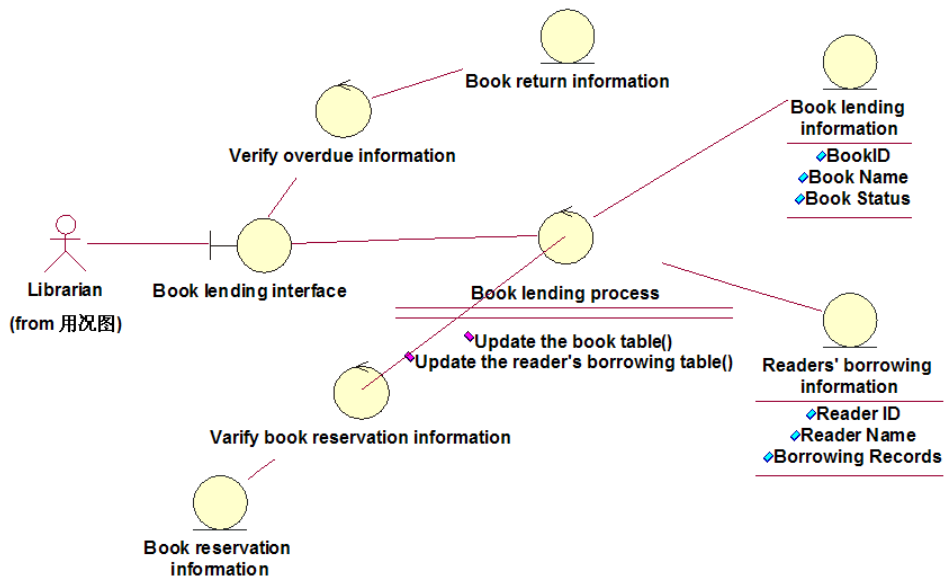Fig. 4. Use case diagram of the library management system

Fig. 5. Analysis chart of the library loan management

3.5. System design of the library management system

3.5.1. Class diagram of the library management system

(1) Reader_class stands for the class of readers. It has many attributes, such as the borrower's ID account (reader_id), Name (reader_Name), Address

136

(Reader_Address), Classes (Reader_classes), the bibliography of the borrowed book (Reader_ bibliography), etc. The most important manipulations are book-borrowing (addborrowed), book-returning (deleteborrowed) and book-reservation (reservation).

(2) Admin_class is the class of the administrators. It has the attrbute of an ID and a Name. The main manipulations are adding, deleting, and updating of readers and books.

(3) Title_class is the class for the bibliography information. It has the attrbutes of book_name, book_author and book_id, etc.

(4) Item_class is the class for a book. It has the attrbute of a book_number. The main manipulations are book reserve (book_reserve), find the book by the title (find_on_title), etc.

(5) Borrow_class is the class for a book's borrowing information. It has the attrbute of the book ISBN (book_ISBN), borrowing date (book_borrow data), etc.

(6) Reservation_class is the class of reservation information. It has the attrbutes of the reservation data (reservation_data), ISBN of the reserved book, ID of the reader for reservation, etc.

(7) Persistent store_class is the class for book's permanent storage. All the activities that relate to the books will use this class.

The interactive relationships between these classes are shown in Fig. 6.

Through the interactive relationship of the different classes in the library management system, we can get the design class diagram of the book lending management shown in Fig. 7.

From the design class diagram we can see that there are eight classes in book lending management. There are three active classes which are the book lending interface active class, the book lending request active class and the book lending process active class. There are five design classes which are book overdue information verification program design class, book return information design class, reservation information verification program design class, book reservation information design class and the borrowing information design class. The three active classes correspond to the three subsystems of the book lending management. The book lending interface active class corresponds to the book lending interface subsystem. The book lending request active class corresponds to the book lending request subsystem. The book lending process active class corresponds to the book lending process system.

First, the librarian does a login of the "book lending interface" active class and gets readers' information through this class. Then the book lending interface active class calls the book lending request active class. The book lending request active class calls the book overdue information verification program design class to judge whether the readers have overdue books. This judgment needs to call the book return information design class to check whether the readers have overdue information by viewing the readers' historical borrowing records. After the validation, the book lending request active class calls the reservation information verification program design class to judge whether the book is reserved by others. This judgment needs to call the book reservation information design class to check

whether this book is reserved by viewing the readers' historical reservation record. Finally, after all the information verification is passed, the book lending request active class calls the book lending process active class. The book lending process active class handles the problem of borrowing books and calls the borrowing information design class. The librarian can update the readers' borrowing information and the book lending information by this design class.
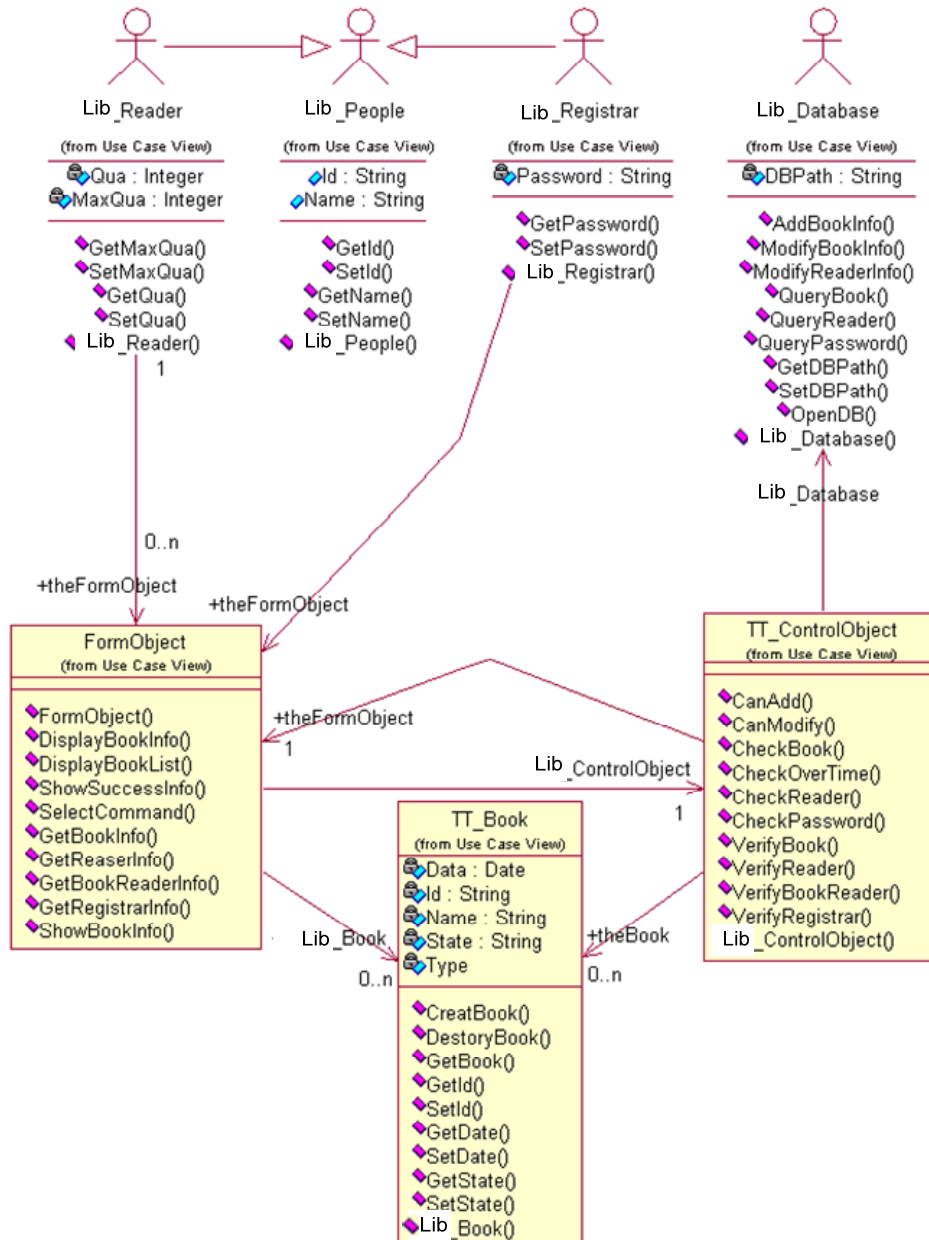

Fig. 6. Interactive relationships between different classes

138
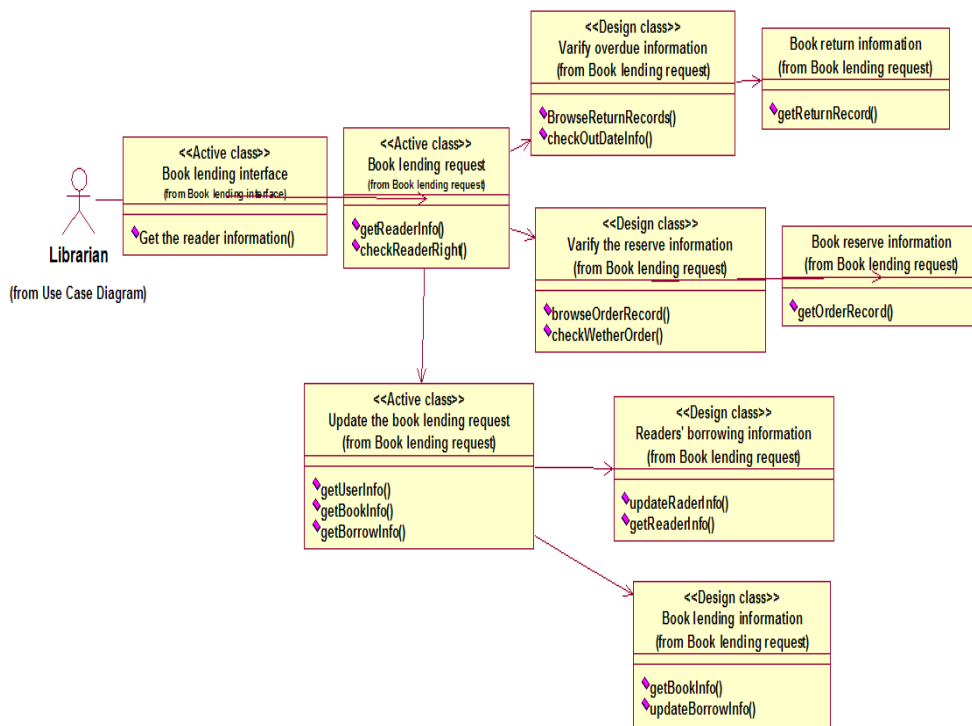
Fig. 7. Design class diagram of book lending management

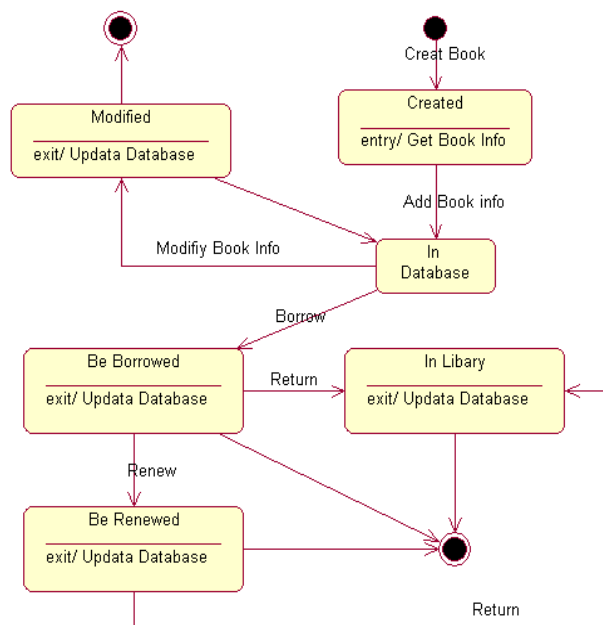The object diagram of the book management system is shown in Fig. 8.

Fig. 8. Object diagram of the book management system

3.5.2. Sequence diagram

According to the design class diagram, we can get the sequence diagram of a book lending management process shown in Fig. 9.
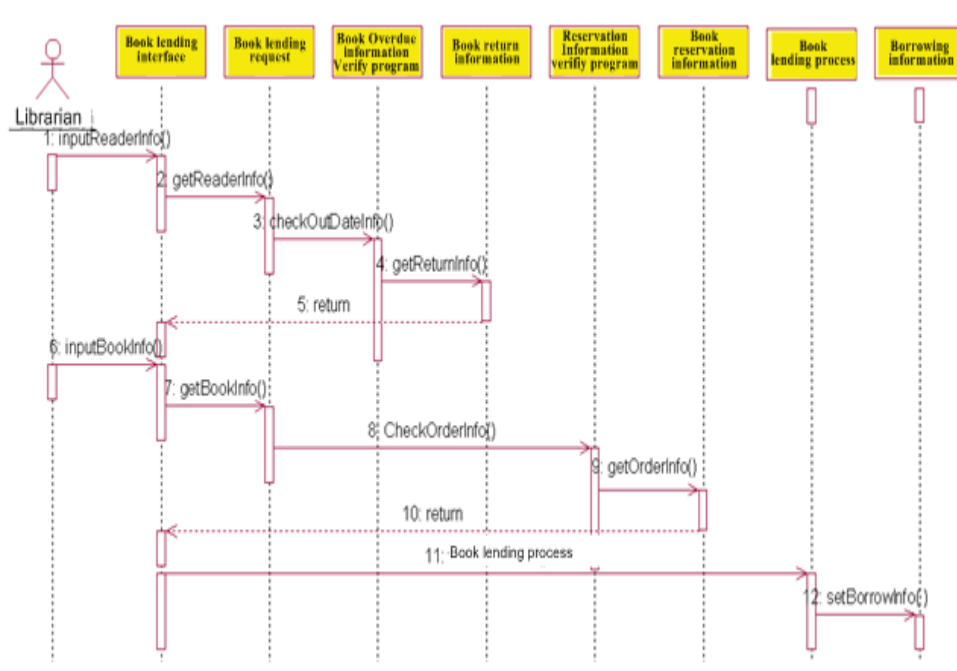

Fig. 9. Sequence diagram of the book lending management

The sequence diagram of the book lending management expresses the following 12 steps.

(1) The librarian logins the book lending interface and inputs the readers' information.

(2) The book lending interface calls the book lending request active class and gets the readers' information.

(3) The book lending request active class calls the overdue information-verifying program and uses this program to judge whether the readers have overdue records.

(4) The overdue information-verifying program calls the book return information design class to check the readers' historical borrowing records and judges whether the readers have overdue books.

(5) After the validation, the information is sent to the book lending interface.

(6) The librarian continues to input the book information into the book lending interface.

(7) The book lending interface calls the book lending request active class and gets the book information.

(8) The book lending request active class calls the reservation information-verifying program and uses this program to judge whether the book is reserved by other readers.

140

(9) The reservation information-verifying program calls the book reservation information design class to check whether this book is reserved by other readers by viewing the book's reservation record.

(10) After the validation, the information is returned to the book lending interface.

(11) The book lending interface active class calls the book lending process active class to handle the book lending problem.

(12) The book lending process active class calls the borrowing information design class to update the readers' borrowing information and the book lending information.

The specific sequence diagrams of Lib_AddInfo, Lib_ModifyInfo, Lib_Borrow, Lib _Return and Lib _Renew are shown in Figs 10-14.
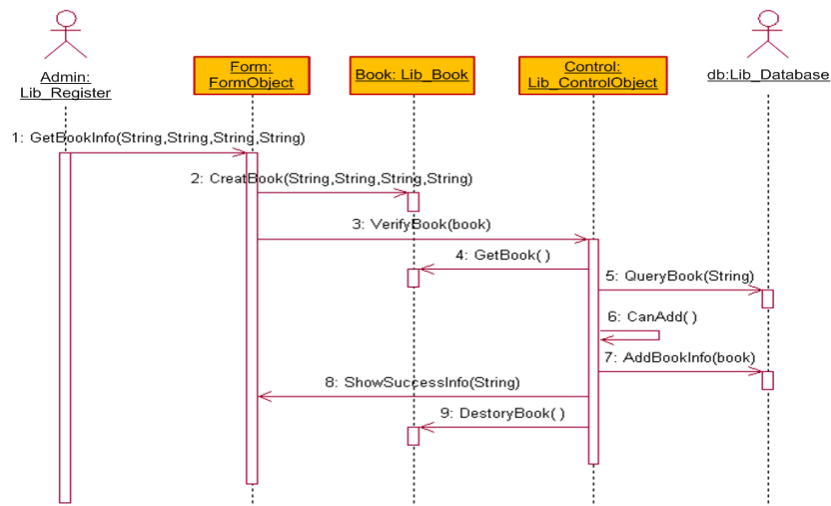

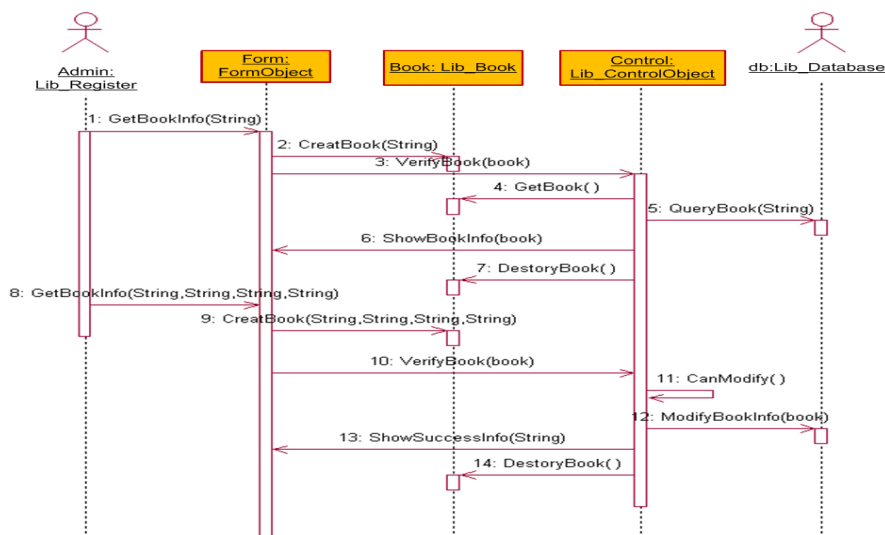
Fig. 10. Sequence diagram of Lib_AddInfo
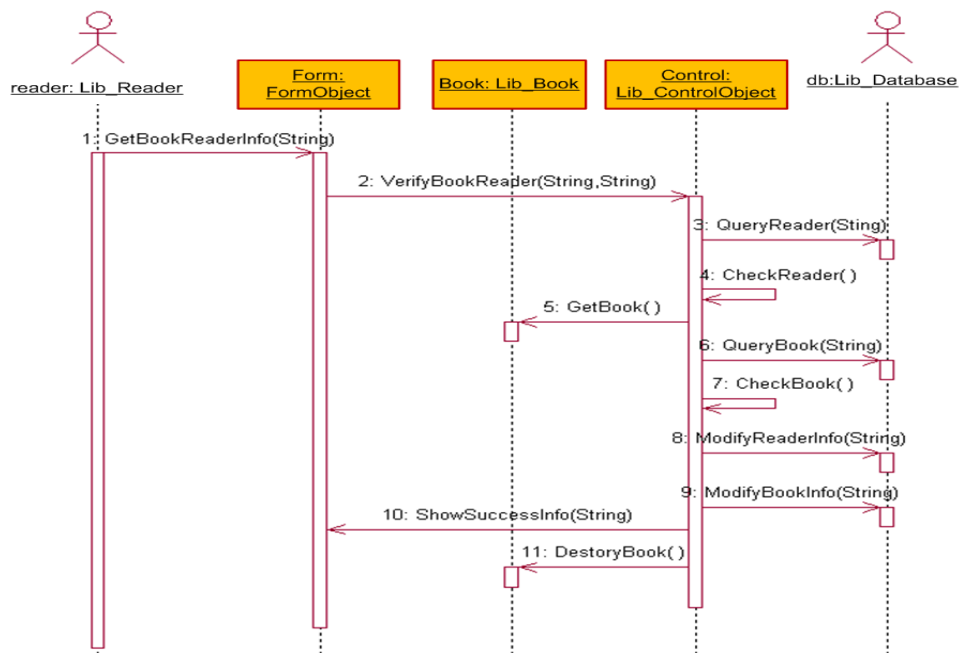

Fig. 11. Sequence diagram of Lib_ ModifyInfo
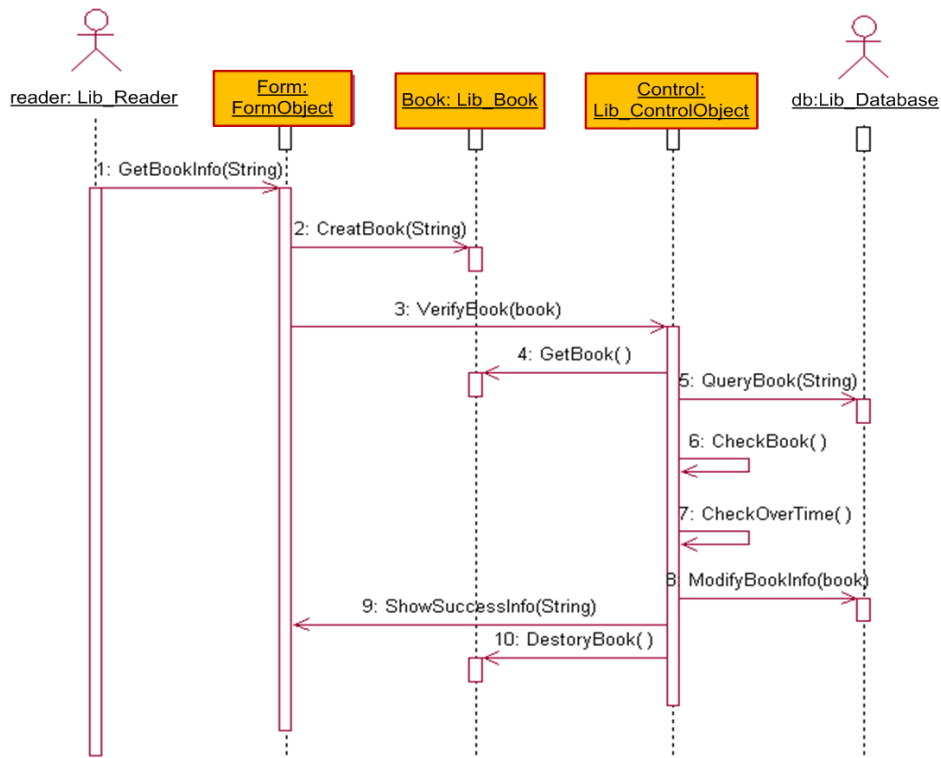
141

Fig. 12. Sequence diagram of Lib_ Borrow

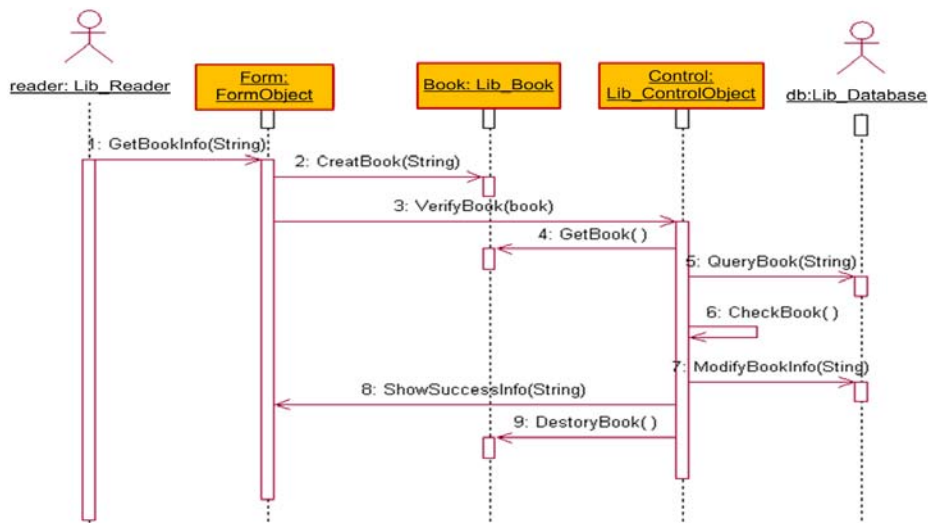Fig. 13. Sequence diagram of Lib_Return

142

Fig. 14. Sequence diagram of Lib_Renew

3.6. Realization of the book management system

After establishment of the system visible model, we can use the rose-based forward engineering to generate the preliminary framework code of the system. Before programming we should select a suitable programming environment, the design language and database, such as the use of Java language and Oracle database. All the database tables, views and stored procedures are designed according to the class relation diagrams of the system database. Finally, we complete the programming of the system application code.

4. Conclusions

The unified modeling language is a powerful, object-oriented and visualized system analysis and modeling language. It uses a set of sophisticated modeling techniques and is widely applied to various areas.

The use of UML-based object-oriented visual modeling can provide the software developers a unified, flexible and understandable representation model, which can reduce blindness in the system design and it is more conducive to the expansion and test of the system. This paper uses the Unified Modeling Language and visualized modeling software tool (Rose) to analyze and design the book management system. A variety of graphics are used to realize the object-oriented analysis, design and development of the entire software system, which shorten the development cycle greatly. Through the design and development of the book management system we can see that, as a modeling language in software engineering, UML represents the development direction of the object-oriented method in software development technology, which has a significant economic value and a very good application prospect.

143

R e f e r e n c e s

1. Y i n g, Z h o u, Z h a n g Y u-h u a. UML and Its Application in the Library Management System. – Journal of Microprocessor, Vol. **5**, 2004, No 1, 52-54.
2. L i, J u n-y i n g. Object-Oriented Unified Modeling Language UML and Its Application. – Journal of Hunan First Normal, Vol. **7**, 2007, No 1, 161-162.
3. L i, W a n g. UML and Software Engineering Fundamentals. Higher Education Press, 2004, 101-115.
4. B o g g s, W., M. B o g g s. UML with Rational Rose 2002 from Entry to the Master. Electronics Industry Press, 2002, 46-277.
5. L i u, H o n g-m e i. UML-Based Library Management System Modeling. – Computer and Modernization, Vol. **3**, 2007, No 1, 100-102.
6. D r u s i n s k y, D., J. B. M i c h a e l, T. W. O t a n i, M.-T. S h i n g. Validating UML Statechart-Based Assertions Libraries for Improved Reliability and Assurance. – In: Proc. of 2nd International Conference on Secure System Integration and Reliability Improvement, 2008, 47-51.
7. L i, T i e-c h e n g. Analysis and Modeling on UML Recommender and Purchase System for Book. – Journal of Tonghua Teachers College, Vol. **28**, 2007, No 4, 77-79.
8. L v, Q i-y i n g. The UML Analysis and Design of the Library Management System for the B/S Model. – China Management Informationization, Vol. **15**, 2012, No 10, 112-114.
9. Z h a o, Q i n g-y a n, Z h a n g C h a o. UML and Its Application in the Design of Electronic Books Online Publishing System. – China Science and Technology Information, 2006, No 4, 44-45.
10. G e, Z h o u. The Application of Unified Modeling Language for Network Book Sale System Development. – Computer and Information Technology, Vol. **15**, 2007, No 2, 46-48.
11. Z h u, C h e n g. Modeling Process of Book Management System Based on UML. – Computer and Modernization, 2011, No 10, 48-50.
12. G a o, L i-p i n g. UML-Based Modeling and Design of Library Management System. – Journal of Zhongzhou University, Vol. **26**, 2009, No 6, 121-123.
13. Z h a n g, L i-j i e, J i a-b i n C h e n, J i n-m e i D u. Design and Implementation of Library Management System Based on UML Models. – Journal of Library and Information Sciences in Agriculture, Vol. **18**, 2006, No 6, 5-7.
14. Y u, Q i a n g. Analysis and Design of Books Management System Based on UML. – Journal of Anhui Technical College of Water Resources and Hydroelectric Power, Vol. **6**, 2006, No 2, 65-67.
15. C h e n, S h i-c h u a n, H u a-n i n L i. Establishment the Analysis Design Model for Books Management System Using UML. – Journal of Shangqiu Vocational and Technical College, Vol. **6**, 2007, No 2, 29-32.
16. C h e n, M i n g. Using UML Language to Model the Library Management System. – Journal of Guizhou University of Technology (Natural Science Edition), Vol. **32**, 2003, No 4, 77-79.
17. C h e n g, C h a n g-p i n, C h e n Q i a n g. Using UML to Construct Library Management System Based on B/S Model. – Journal of Chongqing University of Arts and Sciences (Natural Science Edition), Vol. **29**, 2010, No 3, 42-44.
18. G a o, Z h i-j i a n. Establishment of library Management System Using UML. – Journal of Pingdingshan Institute of Technology, Vol. **14**, 2005, No 2, 49-52.
19. B a i, X i a o-l i n g. The Design of Constructing Library Management System by Applying the Unified Modeling Language. – Journal of Library and Information Sciences in Agriculture, Vol. **21**, 2009, No 12, 113-114.
20. Z h a n g, G u a n g-q u a n, L i u Y a n. Modeling Software Architecture and Their Implement Based on UML of Library Management System. – Journal of Chongqing Teachers College (Natural Science Edition), Vol. **22**, 2005, No 2, 57.