

객체 지향개발 방법론을 이용한 도서관 도서대출관리 시스템 개발

이경구* · 이영수** · 신옥근***

한국해양대학교 IT공학부

Development of Book Rental Management System Based on Object Oriented Methodology

Gyeong Gu Lee · Young Su Lee** · Ok Keun Shin****

Dept. of mechanical and Information Engineering , Korea Maritime University, Busan 606-791, Korea

요 약 : 도서관의 대출관리 업무는 현재 대부분 전산화과정을 통해 통합관리 되고 있지만 소규모 시스템에서는 원격지 프로그램간의 공유자원의 접속성이 항상 보장되어 있지는 않다. 본 연구는 도서 대출관리 시스템을 Client/Server 환경으로 구성하여, 원격지에 위치한 서로 다른 부서의 작업자들이 공유 데이터베이스에 접속하는데 동시성을 보장한다. 또한 설계기법의 표준성을 보장하기 위하여 객체지향개발 방법기법인 Unified Modeling Language Tool을 이용하여 시스템을 설계한다. 더불어 ADO.net 기반의 데이터 연결기법을 이용하여 시스템과 서버의 네트워크 연결과정에서 발생하는 부하를 최소화하도록 구현하였다.

핵심용어 : 도서관 대출관리시스템, Unified Modeling Language, Client/Server, 데이터베이스, ADO.net

ABSTRACT : *Currently, the library rental services are being managed in an integrated manner with the help of computer technologies. However, small-scale rental service programs does not always guarantee the sharing of data resources among co-workers located in remote sites. In this experimental study, emphasis has been put on the concurrent access on shared database in a Client/Server environment so that co-workers can share the resources. The Unified Modeling Language Tool, which is an object oriented development methodology, is adopted to ensure the standardization during the system development. The overhead during the connection to network for server connection is also minimized by using the ADO.net's data coupler technology.*

KEY WORDS : Book Rental Management System, Unified Modeling Language, Client/Server, DataBase, ADO.net

* ilbeum@naver.com

** dudtn14@hanmail.net

*** okshin@hhu.ac.kr

1. 서 론

정보화 사회가 고도화됨에 따라 지식의 중요성이 증대되고 있으며, 컴퓨터와 인터넷통신 기술의 발달에도 불구하고 많은 양의 양질의 지식수요는 도서에 의존하고 있으며 이러한 의존성은 앞으로도 지속될 것으로 예측된다 [1]. 한편 많은 도서수요자들은 필요한 도서를 모두 구입하기 보다는, 도서대출을 담당하는 영리 및 비영리 조직을 통하여 도서수요를 충족하게 된다. 따라서 도서수요의 증대는 도서대출 기관에게 대출업무량 증가, 관리도서권수 증가, 대출업무처리속도 및 업무신뢰도의 보장성 증가 등의 문제를 대두 시키게 되었고 현재는 이러한 문제를 해결하기 위해 도서의 대출 및 관리에 관한 모든 업무가 전산화되어 통합관리 되고 있다.

본 논문은 대학도서관을 모델로 선정하여 도서 대출관리업무의 주요내용을 전산화작업을 통해 통합 구축하되 시스템의 확장용이성 제고 및 설계기법의 표준성을 제공하기 위해 객체지향 개발 방법기법을 적용한다[2]. 이를 위해 시스템 설계 단계에서는 Unified Modeling Language 기법으로 객체지향적 설계방법을 이용한다. 또한 원격지에 위치한 관련 부서들에서 실시간으로 공유 데이터베이스에 접속하여 시스템이 운영되어야 하기 때문에 Client/Server 환경으로 시스템을 구성하여 각 부서들간의 접속의 동시성 및 원격성을 보장하도록 설계한다. 본 연구를 진행하기 위하여 대학도서관의 주요업무내용과 사용자의 요구사항을 설명한 후, 시스템에 대한 데이터베이스와 UML 기법을 이용한 설계내용을 소개하며, 시스템 구현에 관한 주요내용들을 정리하였다. 아울러 주어진 요구사항에 적합한 시스템을 개발 후 시험하였다.

2. 시스템의 요구분석

2.1 도서관 주요업무 처리과정

학생이 도서관을 통해서 신규도서를 신청하고 소장된 도서를 대출하는 과정은 다음과 같이 구성된다. 도서관에 소장되어 있지 않거나 많은 수요성을 가지는 도서는 학생이 대출실과 수서과를 통해서 서면으로 신규도서신청을 할 수 있다. 학생으로부터 신규 신청된 정보는 수서과의 수서내역에 기록되어 관리되며 신규도서에 대한 구입여부를 검증한 다음 구입승인이 이루어지면 요구된 도서를 거래처를 통해서 주문한다. 신규도서가 도서관에 도착하면 도서의 정보를 도서관리 대장에 등재 후 도서에 특정규칙을 따르는 일련번호를 부여하고 정리한 다음 대출실에 비치되게 된다.

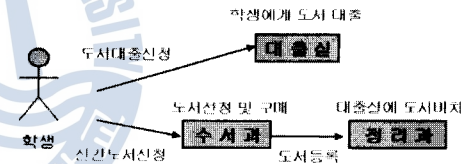


Fig. 1 Book requisition and rental process

도서를 대출하는 과정은 학생이 대출실에서 대출 희망 도서를 직접 선정하여 대출실 직원에게 해당도서에 대한 대출요청을 구두로 통보함과 동시에 대출희망 도서와 학생의 정보가 기록된 학생증을 전달하면 대출실직원이 대출가능 여부를 검증 후 학생에게 해당도서를 대출하게 된다. Fig. 1에 학생의 도서대출과 신간도서신청 과정을 도식화 하였다.

2.2 사용자의 기능적 요구사항

시스템 요구분석 단계에서 조사된 사용자의 요구사항들을 객체지향 개발방법론을 이용하여 Fig. 2의 UseCase Diagram으로 구성하였다. 하나의 단위 원은 기능적 정체성을 보유한

고유의 사용사례이며 연결선의 구분에 따라 지정된 사용자 혹은 공용의 사용자가 참조 할 수 있는 구조이다. 시스템의 주요 사용자는 수서과와 대출실의 직원으로 구성되며 연결선을 통해서 기능적 관계도를 형성한다. 요구분석 과정을 통해 도출된 사용사례들은 각 기능간의 응집도를 높이는 동시에 전체 기능들 간의 유기적 결합도를 최소화 하는 방식으로 설계되어 객체지향 개념에 입각한 설계방법론의 모델이 되었다[2]. 다시 말해 설계기법의 표준화를 통해서 향후 프로그램의 재사용성 증대 및 유지보수의 용이성을 보장받게 되었다. Fig. 2 의 도식화된 UseCase Diagram의 세부적인 설명은 Table 1에 나타난다. 시스템의 주요 사용자인 수서과와 대출실 직원들은 각 자에게 요구된 기능들을 사용하여 도서대출에 관련된 업무를 처리하게 된다. 또한 사용자 구분없이 제공되는 기능들을 통하여 공유 데이터베이스에 대한 실시간접속성이 보장되므로 공유자원의 활용도를 높이는데 기여하게 된다. 수서과 직원에게만 제공된 기능은 두 가지로 거래처에서 구매한 신규도서의 정보를 입력하는 “신규도서등록” 기능이 포함되며 학생이 신간도서 구입 요청시 해당도서가 도서관에 소장된 여부를 확인하기 위한 목적으로 “신청도서 보유확인” 기능을 추가 한다. 시스템의 주요기능인 대출관리의 업무를 수행하는 대출실직원은 해당학생에게 도서를 대출해주며 대출된 도서를 반납받는 기능을 가진다.

3. 시스템의 설계

3.1 데이터베이스 설계

본 시스템은 Client/Server환경으로 구현되는 시스템으로 중앙의 데이터베이스 서버에 저장된 공유 데이터베이스를 원격에 위치한 여러 부서에서 동시에 접속한다.

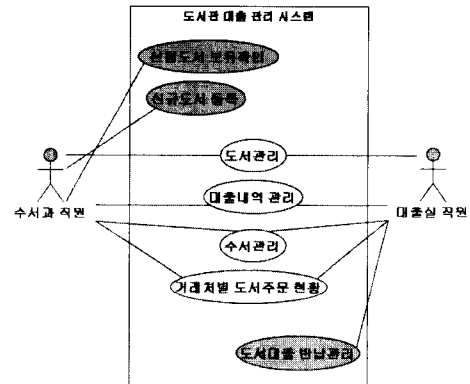


Fig. 2 Usecase diagram

Table 1 User requirements

부서 분류	요구 사항	내 용
수서과	신규도서 등록	학생이 대출실에 신청한 신규도서가 도서관에 도착하면 신청 도서의 정보를 입력
	신청도서 보유 확인	신규도서를 신청하기 이전, 도서관의 소장자료 여부를 확인
대출실	도서 대출	대출을 희망하는 도서의 정보를 대출내역에 기록 후 학생에게 대출
	도서 반납	학생으로부터 도서가 대출되면 해당도서의 대출내역을 갱신 후 도서를 수령
공통사항	수서 관리	학생이 신청한 신규도서에 관한 정보인 수서내역을 관리
	거래처별 도서주문 현황	등록된 도서정보를 이용하여 거래처를 기준으로 도서를 그룹화하여 거래처별 도서리스트를 작성
	도서 관리	도서관에서 소장 중인 해당 도서에 관한 세부적인 정보들을 기록하여 도서를 관리
	대출내역 관리	도서 및 대출에 관한 정보를 포함하는 대출내역 (어떤 학생이 무슨책을 언제 대출하여 언제까지 반납일이었는데 현재는 어떤 상태이며 반납을 완료했다면 몇일에 반납원료를 했는지 여부) 을 종합적으로 확인 관리

따라서 접속에 대한 동시성 및 원격성을 보장해야만 한다. 만약 이 사항이 충족되지 않게 된다면, A부서에서 변경한 내용을 곧바로 B부서에서 참조하였을 경우 B부서는 변경된 최신 정보를 참조하지 못하며 A부서가 변경하기 이전의 자료를 참조하게 되어 모순된 정보를 인식하는 문제가 발생하

게 된다. 본 연구는 이러한 문제를 해결하기 위해 ADO.net 의 비연결형 서버접속 기법을 근간으로 구성하였다. 테이블은 제 3 정규화 과정을 통해 생성된 네 개의 테이블로 구성되며 각 테이블들 간의 기본키 및 외래키가 유기적인 종속관계를 형성한다. 테이블 설계 시 시스템의 확장성을 고려하여 시스템과 데이터베이스 연동 시 발생하는 트랜잭션의 복잡성을 최소화 시키면서도 각 테이블이 가질 수 있는 정보의 이상화(Anomy)현상을 최소화 시키는데 주안점을 두었다[3]. 이 관계는 서로 대립되는 관계로서 테이블이 저장하는 정보를 최소화 시키면서 이상화 현상을 제거할 경우, 시스템과의 연동 시 요구되는 트랜잭션이 복잡해지는 구조가 일반적이다[4]. 이 복잡성을 해소하기 위하여 시스템의 기능적 요구사항의 설계과정에서 객체지향 개발방법론에 근거한 소프트웨어 공학의 개념을 적용하여, 개별 기능이 가지는 기능적 응집도는 높이고 각 기능들 간의 유기적 결합도를 낮추는 기본설계에 중점을 두었다. 때문에 개별기능들의 속성이 보다 명확하게 정의될 수 있었고, 이를 근간으로 데이터베이스의 설계가 진행되었기 때문에 시스템의 트랜잭션의 복잡성을 감소시키면서도 테이블의 구조를 간소화 시키는 긍정적 결과를 유도할 수 있었다. 데이터베이스를 설계하는 과정 중 개념적 설계 단계에서 생성된 Fig. 3의 ERD (Entity-Relation Diagram)는 Entity의 구성과 유기적 연관관계를 나타낸다. 도식화된 기호들을 통해서 각 개체의 기본키(Primary Key)와 외래키(Foreign Key)를 구분하며 개체간의 관계가 연결선의 종단에 표시되어 있다. "book" 과 "rental"의 관계는 1 : 0..N의 관계를 형성하여 "하나의 도서는 0개 혹은 복수개의 대출내역을 가진다" 라는 조건을 만족하며 동일한 방식으로 나머지 개체들 간의 종속관계를 확인할 수 있다. Table 2

는 개념적 설계단계에서 구현된 테이블의 세부적 속성을 나타낸다.

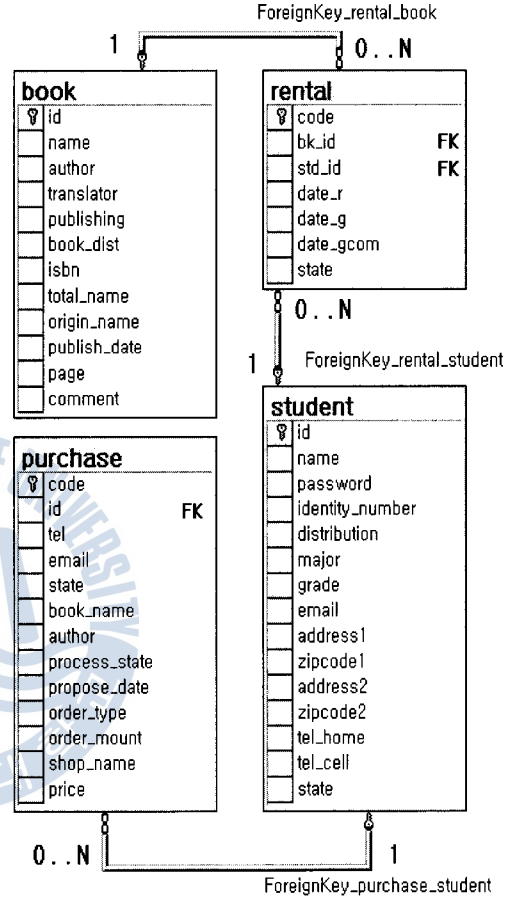


Fig. 3 Entity Relationship Diagram

Table 2 Attributes of database tables

Attribute	Data Type	Notation
book		
id	char	Primary Key
name	varchar	도서명
author	varchar	저자명
translator	varchar	번역자
publishing	varchar	출판사
book_dist	varchar	도서코드
isbn	varchar	ISBN 부여번호
total_name	varchar	총서명

origin_name	varchar	원서명
publish_date	char	출판일자
page	int	페이지수
comment	varchar	서평
purchase		
code	int	Primary Key
id	char	student의 Foreign Key
tel	text	전화번호
email	varchar	이메일주소
state	varchar	신분(학부생, 대학원생)
book_name	varchar	도서명
author	varchar	저자명
process_state	varchar	처리상태
propose_date	varchar	처리완료 일자
order_type	varchar	주문형태
order_mount	varchar	주문수량
shop_name	varchar	거래처명
price	money	가격
rental		
code	int	Primary Key
bk_id	char	book의 Foreign Key
std_id	char	student의 Foreign Key
date_r	char	도서 대출일자
date_g	char	도서 반납 기한일자
date_gcom	char	도서 반납 완료 일자
state	varchar	현재 처리 상태 (대출중, 반납)
student		
id	char	Primary Key
name	varchar	성명
password	varchar	비밀번호
identity_number	varchar	주민등록번호
distribution	varchar	학부구분
major	varchar	학과구분
grade	varchar	학년
email	varchar	이메일주소
address1	varchar	주소1
zipcode1	text	우편번호1
address2	varchar	주소2
zipcode2	text	우편번호2
tel_home	text	집전화번호
tel_cell	text	휴대폰번호
state	varchar	현재상태(재학, 휴학, 재적)

준으로 순차적으로 진행된다. 즉 시스템의 관점에서는 단위 기능들의 정의를 객체지향 방법론의 개념을 적용하여 단위 객체의 응집도는 높이며 객체간의 유기적 결합도는 낮추도록 설계하였다[2]. 단위 객체의 설계 및 구현은 해당 객체의 정의 및 구현의 독립성을 높이도록 설계하는 반면 해당 객체의 내부적 구현방법은 Fig. 4에 나타난 것과 같이, 정의된 함수 및 메시지가 시간을 기준으로 순차적으로 진행된다. 즉 객체의 정의는 독립성을 보장해야 하지만 그 내부 구현은 시간적 순서를 기준으로 절차적 방법으로 구현되는 것이다.

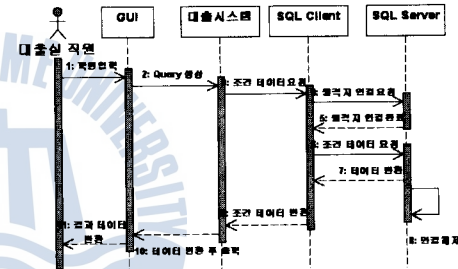


Fig. 4 Sequence Diagram

Fig. 4는 대출실 직원이 학생에게 도서를 대출하기 위해 대출시스템을 통하여 학생의 학번을 입력한 후의 처리과정을 순차적으로 나타낸다. 대출실 직원은 사용자 프로그램을 통해서 특정 학생의 학번을 입력하면 현재 프로그램에 설정된 정보에 적합한 SQL Query를 생성하여 PC에 설치된 SQL Client에게 데이터를 요청한다. SQL Client는 원격지에 위치한 SQL Server를 LAN을 통해서 접속하여 요구된 정보를 요청한 후 정확한 결과값을 반환 받게 된다. 다음으로 대출시스템은 반환된 결과를 ADO.net의 DataSet의 형식에 적합하도록 XML형식으로 변환 한 다음 사용자 프로그램에 정보를 최종적으로 출력한다. 학생의 학번을 대출시스템에 입력하여 원격지에 위치한 SQL Server로부터 신상정보와 도서대출내

3.2 Sequence Diagram Design

UML을 통하여 작성된 Sequence Diagram은 객체의 집합, 또는 액터가 교환하는 메시지의 교환이 시간적흐름을 기준으로 순차적으로 나타난다[2]. 사용자 액터가 특정한 결과를 반환받기 위해 이벤트를 발생시키면 해당기능의 구현이 정의된 각 단계들이 시간의 흐름을 기

역의 정보를 반환 받는 과정처럼 나머지 객체들의 구성 또한 내부적으로는 절차적으로 진행된다.

```

cndbid = New SqlCommandBuilder(ad) //신규 어댑터연결
ad.Fill(Ds1, "rental") // DataSet을 어댑터에 추가
DataGridDisplay() //DataGrid 출력할수 호출
conn.Close() //모든 작업수행후 접속해제
End Sub
    
```

4. 시스템의 구성

4.1 시스템과 데이터베이스 연결

구현된 시스템은 ADO.net 기반의 비연결형으로 운용되는 시스템으로써, 최초 데이터를 가져올 경우에만 원격지 서버에 접속하며, 서버로부터 조건 데이터를 전달받게 되면 서버와의 연결을 단절하여 시스템에 가해지는 부하를 최소화하도록 구현된다.

ADO.net의 주요한 특성은 시스템과 데이터베이스 사이에 원본 데이터베이스의 특정 부분을 복사한 DataSet이 존재하여, 가장 빈번하게 발생하는 검색작업일 경우 시스템이 데이터베이스에 접속하지 않고 이미 생성해 놓은 DataSet에서 데이터를 가져오는 구조이기 때문에 시스템과 데이터베이스 서버에 발생하는 부하가 현저히 감소되는 구조이다 [6]. 본 시스템 또한 출력된 정보를 특정 키워드를 기준으로 필터링 하여 정보를 재정렬할 경우에는 위의 기술을 적용하여 시스템을 구현하였다. Table 3은 단위 프로그램이 원격지에 위치한 데이터베이스 서버에 접속하여 데이터를 가져온 후 데이터그리드에 출력하고 연결을 종료하는 과정을 예시한다.

Table 3 An example of database connection and release

```

Private Sub rental_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load //정의부
conn.ConnectionString = "server = " & MD_Server_name & ";UID=" & MD_UID & ";PWD=" & MD_PWD & ";DATABASE=" & MD_DataBase //연결스트링
conn.Open() //초기 연결객체 생성
sql = "select * from rental" //SQL 문 생성
cmd = New SqlClient.SqlCommand(sql, conn) //명령개체생성
cmd.CommandType = CommandType.Text //명령개체형식정의
ad.SelectCommand = cmd //데이터 어댑터에 명령적용
    
```

4.2 시스템의 구조

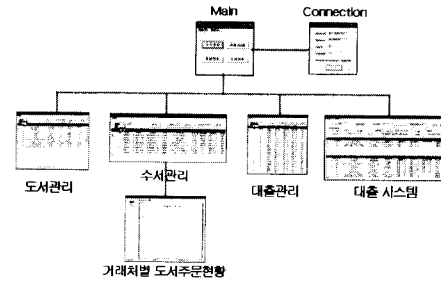


Fig. 5 System Architecture

시스템은 각각의 고유기능으로 구현된 프로그램들이 유기적으로 연동되어 구성된다. Fig. 5의 내용처럼 프로그램은 사용자의 접근순서에 따라 계층적인 구조를 형성하므로 상위단계를 거쳐야만 하위단계의 프로그램으로 진입할 수 있다. 일반적으로 시스템을 구성하는 여러개의 단위프로그램들은 하나의 레이아웃에 통합되어 운용되지만 본 시스템은 각 프로그램의 고유한 속성이 강하며 프로그램간의 유기적 연관성이 부족한 이유로 Fig. 5와 같이 단위 프로그램의 독립성을 보장하는 계층적인 구조로 구성하였다. 최상위 Main 프로그램의 데이터베이스 연결을 시도한 후부터는 단위 프로그램이 다른 프로그램의 도움없이 독자적으로 구현될 수 있는 프로세스들을 개별적으로 모두 포함하도록 구성되었다.

4.3 데이터베이스 연결 프로그램

시스템은 원격에 위치한 프로그램이 중앙의 데이터베이스 서버에 접속하여 구현되는 Client/Server환경이다. 따라서 시스템을 구동

하기 위해서는 시스템 시작과 동시에 원격지에 위치한 데이터베이스 서버에 접속되는 과정이 반드시 수반되어야 한다. 만약 이 과정에서 특정한 문제가 발생하였다면, 시스템은 데이터베이스 서버에 접속실패 상태로 전이되어 시스템을 구동할 수 없게 되므로 반드시 시스템 구동의 초기 과정으로 데이터베이스 서버에 정확히 접속하여야만 한다. 원격지에 접속하기 위해서는 SQL Client에서 제공되는 Connection Domain을 Fig. 5의 Connection 프로그램에 입력하여 접속하며 접속이 정확하게 진행된다면 SQL Server의 Version을 반환받아 사용자 화면에 표시하며 알림창을 통하여 접속이 정확하게 완료되었음을 통보받게 되어 시스템은 정상 대기상태로 진입하게 된다.

4.4 수서관리 프로그램

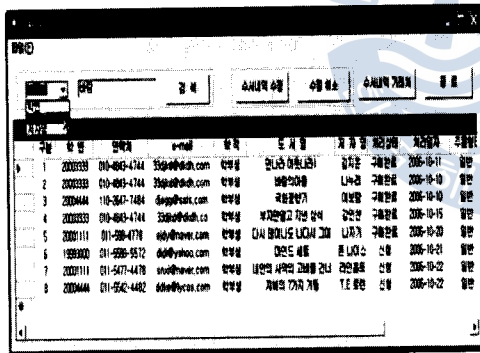


Fig. 6 The user interface of the purchase management

Fig. 6의 수서관리 프로그램은 학생이 신청한 신간도서를 구입 심사하여 구매가 승인된 수서관리내역들의 정보를 관리한다. 기존에 등록된 정보들을 특정기준(학번, 도서명, 저자명)으로 선정 후 입력한 키워드를 기준으로 데이터를 필터링하여 조건에 적합한 데이터만 표시하는 기능을 제공하며 표시된 데이터를 직접 변경하여 저장하는 기능을 제공한다. 또한 수서관리

역에 표시되는 행들은 상이한 거래처 정보를 가지는데, 거래처를 기준으로 그룹화하여 하위 프로그램인 거래처별 도서리스트를 확인할 수 있도록 구현하였다.

4.5 대출관리 시스템

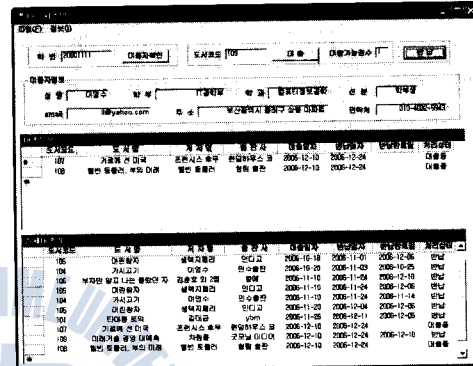


Fig. 7 The user interface of the book rental management

Fig. 7에 본 시스템의 핵심은 대출관리 시스템을 보인다. 주요기능은 학생에게 도서를 대출하면서 시스템에 도서대출정보를 기록하며 학생이 도서를 반납하면 반납한 도서에 대해서 대출내역을 반납에 적절하게 수정하는 시스템이다. 학생이 도서 대출을 신청하면 학생으로부터 학생증과 도서를 전달받아 프로그램 상에 학생의 학번을 입력한다. 프로그램은 학생에 대한 개괄적인 학적사항을 표시해 주며 상위 데이터그리드에는 현재 대출중인 도서의 목록이 표시하고 하위 데이터 그리드에는 해당학생이 그동안 도서관에서 대출한 모든 내역들을 표시하게 된다. 대출하고자 하는 도서의 도서코드를 도서코드란에 입력 후 대출명령을 가하면 해당도서가 학생에게 대출승인이 발생하여 이 모든 과정이 데이터베이스에 기록되어 대출과정이 완료되게 된다. 반납과정은 해당학생의 학번을 입력한 후 반납할 도서의 도서코드를 입력하고 반납명령을 가하면 도서

대출내역에 저장된 해당학생의 대출거래내역의 대출정보가 반납으로 변경되고 반납일자가 기재되면서 반납과정이 완료되게 된다. 대출시스템을 이용하여 도서를 대출할 경우 학생의 신분에 따라서 차등적으로 도서 대출가능권을 설정해야만 한다. 즉, 학부생에게는 최대 대출가능권을 3권으로 설정하는 반면 대학원생에게는 5권으로 설정하는 과정이 수반된다. Fig. 7의 예는 신분이 학부생이며 전체 대출가능권수인 3권에서 2권의 도서를 대출하였기 때문에 우측상단에 표시된 대출가능권수가 '1'로 표시되며 대출권수도 1권으로 제한되게 된다. 대출시스템의 내부적 동작 방법은 다음과 같이 구성된다. 최초 프로그램이 실행된 상태에서는 해당학생의 정보를 미입력한 상태이기 때문에 사용자 정보, 대출목록, 전체대출기록란에는 모두 공란으로 표시된다. 대출실직원이 학번을 입력하여 사용자확인 명령을 가하면 Fig. 4의 절차적 과정을 거쳐서 원격지 데이터베이스 서버로부터 이용자에 대한 신상 및 도서정보를 반환 받게 되어 사용자정보와 대출목록, 전체대출기록에 데이터를 출력해주게 된다. 또한 특정도서코드를 입력하여 도서를 대출하는 과정은 학번과 도서코드가 키워드가 되어서 SQL문을 생성하고 원격지 데이터베이스(Fig. 3의 'rental' Table)에 새로운 내용을 추가함으로써 대출에 대한 모든 과정이 완료된다. 다음으로 데이터베이스 갱신 후 변경된 내용을 실시간으로 사용자에게 표시를 해주어야 하기 때문에 새로운 연결작업을 생성하여 해당학생의 변경된 정보를 다시 읽어 와서 사용자정보, 대출목록, 전체대출기록에 다시 출력해주는 과정이 수반된다. 도서를 반납하는 과정 또한 도서대출 과정과 동일한 절차로 수행되지만 SQL문의 내용이 대출의 경우에는 신규행을 생성하는 문장인 반면 반납의 경우는 이미 생성된 대출정보를 Update

Query문을 통하여 Fig. 4의 'rental' Table의 date_gcom, state열을 수정함으로써 반납의 절차가 이루어지게 된다. 프로그램상에 데이터를 표현 하는 방법은 학생정보 출력, 대출과정, 반납과정이 모두 동일한 절차를 통해서 표현된다. 만약 특정 트랜잭션을 통해서 반환값을 전달 받았다면 프로그램상에 기존에 출력되어 있던 모든 정보들을 삭제한 후 전달받은 결과값을 출력해주는 방법으로 구성하였다.

Fig. 5의 시스템 구성도에 포함되는 도서관 관리, 대출관리, 수서관리의 주요 작업들은 이미 가져온 데이터를 검색하여 정보를 확인하는 과정들로 구성되기 때문에 최초 한번의 연결만을 통해서도 재연결의 추가 과정없이 프로그램이 운용되어 비연결형 구조에 적합한 구조성을 가지게 된다. 반면 대출시스템의 경우, 학생의 학번을 입력하여 조회할 때마다 해당 조건에 적합한 SQL문을 생성하여 데이터베이스에 지속적으로 연결해야 하며 대출 및 반납의 과정이 진행되는 과정 또한 SQL문을 생성하여 단위 트랜잭션에 대한 연결을 수행해야 한다. 따라서 대출시스템의 운용은 다른 프로그램에 비해서 더욱 빈번한 접속을 시도하며 더 많은 자원을 요구하게 된다.

5. 시스템연결과 원격 접속시험

5.1 시스템 연결 구조도

데이터베이스의 원격접속성을 보장하기 위해서 Microsoft SQL Server를 데이터베이스 서버로 사용하며 Client측에서는 SQL Client를 중개자로 사용하여 시스템을 구성한다[5]. 본 시스템이 원격지에 위치한 데이터베이스에 연결이 가능한 이유는 SQL Client와 SQL Server와의 유기적 연결의 도움을 받아서 손쉽게 가능하게 된다. 따라

서 시스템 구현과정에는 SQL Client와 SQL Server의 설치 및 네트워크상의 설정 단계가 반드시 포함되어야 한다. Client측의 시스템의 연결을 Fig. 8에 보인다.

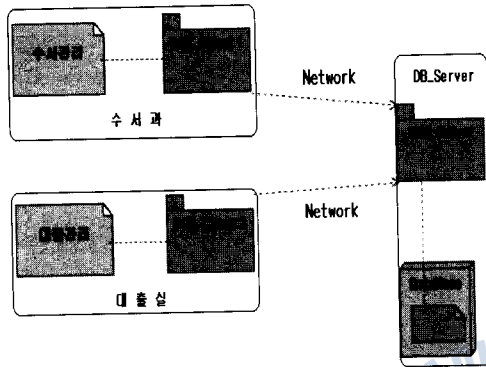


Fig. 8 Network Architecture

5. 2 시스템의 원격 접속시험

SQL Server와 SQL Client간의 연결이 유기적으로 동작 하도록 시스템 간의 네트워크 인식에 관한 설정과정을 거친 후 해당 SQL Client의 Domain Name 을 확인하여 시스템의 초기 연결 프로그램을 통해서 데이터베이스에 접속한다. 접속이 정상적으로 수행되면 접속과정에 대한 성공적인 메시지를 반환 받는 반면, 특정문제 (도메인네임 부정확, 네트워크이상, 데이터베이스 서버이상) 에 직면하게 될 경우 접속이 완료되지 못했다는 메시지를 통보 받게 될 것이다. 본 시스템의 데이터베이스 접속과정은 원격지에 위치한 데이터베이스 서버에 시스템이 직접적으로 접근할 수 없다. 시스템 내부에서는 SQL Client의 Domain Name 만을 필요로 하며, 얻어진 Domain Name을 ADO.net 기반의 Connection 기술을 이용하여 간단하게 연결하는 구조를 채택했다. 따라서 시스템의 구현과정을 간소화시킴과 동시에 프로그램 모듈의 감소를 통해서 Client측의 부하를 줄이는

구조를 가지게 되었다.

6. 결 론

대학도서관에서 학생에게 도서를 대출 및 반납하기 위해서 필요한 요구기능들을 충족하는 소규모 시스템을 구현 하였다. 시스템 구축에 필요한 데이터베이스의 설계 및 구축 과정을 수행하였으며 시스템의 기능적 요구사항을 충족하는 프로그램을 구현하고 시험하였다.

본 시스템의 구축을 통해서 원격에 위치한 수서과 및 대출실 직원들은 도서관업무에 관련된 정보를 실시간으로 갱신된 최신의 상태로 참조할 수 있게 되었으며 학생에게 대출한 도서자료와 대출내역들의 전산화 관리를 통해서 업무의 신뢰성과 신속성을 향상시키게 되었다. 설계단계에서 적용된 객체지향개발 방법론을 통해서 설계의 표준성을 제고하였으며 프로그램의 재사용성을 높이며 향후 시스템 확장을 용이하게 되었다. 향후 과제로는 대학내부에서 구현되는 다른 시스템의 데이터베이스를 본 연구를 통해 생성된 데이터베이스와 통합하여 운영하는 과정에서 발생하는 정보의 중복성 문제를 해결하는 것이며, 단위 프로그램이 보유하는 기능들의 특성을 더욱 더 세분화시킬 필요성 있다. 이를 구현하기 위해서는 보다 엄격한 기준의 요구사항조사단계 및 설계과정이 진행되어야 할 것이다.

참 고 문 헌

- [1] 이미향, [논문] 도서관 서비스 품질의 구성요인에 대한 이용자 인식연구, 성균관대학교, 2004.
- [2] 최은만, 객체지향 소프트웨어 공학, 사이텍

미디어, 2005.

- [3] 김완호, 초급 프로그래머를 위한 Microsoft SQL Server 2000, 정일, 2002.
- [4] 이춘식, 업무 영역별 데이터베이스 설계와 구축, 정보문화사, 2002.
- [5] <http://www.microsoft.com/Korea/MSDN/vbasic/default.aspx>, MSDN
- [6] Matthew Borthiker의 1명, Professional Visual Basic.NET Transactions, 정보문화사, 2002.

원고접수일 : 2007년 1월 4일

원고채택일 : 2007년 1월 24일

