사물인터넷융복합논문지 Vol. 12, No. 5, pp. 149-156, 2026 https://doi.org/10.20465/KIOTS.2026.12.5.149

온라인 쇼핑몰 시스템 설계 및 구현에 대한 심층적 고찰

공기식*
남서울대학교 교수

An In-Depth Analysis on the Design and Implementation of an Online Shopping Mall System

Ki-Sik Kong*
Professor, Department of Multimedia

요 약 최근 비대면 경제 생태계의 확대와 스마트 커머스 시장의 가속화에 따라 온라인 쇼핑몰 사이트 구축 및 운영은 기업 생존의 필수 요소로 자리 잡고 있으나, 신규 웹 개발 인력의 실무 업무 수행 능력과 데이터베이스 연동 역량은 산업계의 요구 수준에 미치지 못하고 있는 실정이다. 본 논문에서는 온라인 쇼핑몰 사이트를 체계적으로 구축하기 위한 주요 모듈별 프로세스 흐름, 데이터베이스 모델링, 그리고 핵심 알고리즘들을 분석 및 제시한다. 특히, '즉시 구매' 처리를 위해 단기 생명주기 기반의 '임시 장바구니(tempcart) 개념'을 도입하여, 기존 장바구니 구현 시에 사용되었던 소스 코드들을 상당부분 그대로 재활용함과 동시에, 수많은 다중 사용자들의 동시 '즉시 구매' 시에도 웹 애플리케이션 서버(Web Application Server)의 메모리 과부하를 원천 차단할 수 있는 효율적인 설계 방안을 제안하였다. 또한, Apache JMeter를 활용한 부하 테스트를 통해 최대 2,500건의 동시 트랜잭션 환경에서 에러율 0.00% 및 평균 응답 속도 0.9~1.4ms의 고속 처리 성능을 실증하였다. 본 연구의 결과는 다양한 웹 개발 환경에서도 보편적 확장성이 있으므로, 신규 웹 개발 인력에게 표준화된 실무 가이드라인을 제공하는 데 기여할 수 있을 것으로 기대된다.

주제어 : 온라인 쇼핑몰, 인터넷 쇼핑몰, 전자상거래사이트, 웹 애플리케이션, 데이터베이스 모델링

Abstract With the recent expansion of the non-face-to-face economic ecosystem and the rapid growth of the smart commerce market, building and operating online shopping malls has become essential for corporate survival; however, the practical work performance and database integration competencies of new web developers often fall short of industry expectations. This paper analyzes and presents the process flows, database modeling, and core algorithms for key modules required to systematically build an online shopping mall. In particular, we introduced the concept of a 'temporary cart (tempcart)' based on a short lifecycle to handle 'instant purchases,' allowing us to reuse a large part of the source code from the existing cart implementation while also proposing an efficient design that can completely prevent memory overload on the Web Application Server when many users make 'instant purchases' at the same time. Furthermore, load testing using Apache JMeter empirically demonstrates high-speed processing performance, achieving a 0.00% error rate and an average response time of 0.9–1.4 ms under a concurrent load of up to 2,500 transactions. The results of this study are universally extensible to various web development environments, and are thus expected to contribute to providing standardized practical guidelines for new web developers and learners.

Key Words : Online Shopping Mall, Internet Shopping Mall, E-commerce Site, Web Application, Database Modeling

*교신저자 : 공기식(kisik.kong@gmail.com)
접수일 2026년 09월 30일 수정일 2026년 10월 15일 심사완료일 2026년 10월 19일

www.kci.go.kr

1. 서론

정보통신기술의 급격한 발전과 더불어 온라인 쇼핑몰 사이트는 스마트 디바이스들과의 결합을 통하여 현대 사회에서의 핵심적인 유통 채널로 자리 잡아 왔다[1, 2]. 또한, 최근의 급격히 늘어나는 1인 가구의 증가 및 코로나 19 이후 비대면 경제 생태계의 확대 경향에 따라, 전자상거래 사이트 구축 및 운영은 기업의 생존을 결정지을 수 있는 필수적인 요소로 자리매김해 가고 있으며, 이에 따른 웹 및 모바일 앱 개발 인력에 대한 산업계의 수요는 점점 더 증가해 가고 있는 추세이다[3]. 그러나 이러한 시장 트렌드와 다양한 기술적 요구사항의 급증에도 불구하고, 현장에서는 전문성을 갖춘 숙련된 웹 개발자의 부족 현상이 지속되고 있으며, 신규 대졸 웹 개발 인력의 실무 업무 수행 능력과 시스템 이해도는 산업계가 요구하는 수준에 미치지 못하고 있는 것이 현실이다[4]. 특히 기초적인 CRUD(Create, Read, Update, Delete) 연동부터 세션 기반의 사용자 인증, 구매 로직의 데이터베이스 모델링에 이르기까지의 전반적인 프로세스를 체계적으로 이해하고 통합 설계할 수 있는 실무적인 가이드라인이 대학 차원의 교육에서는 여전히 잘 이루어지지 못하고 있는 것이 현실이다. 최근의 웹 응용 프로그램 개발 환경은 다양한 클라우드 프레임워크로 다변화되고 있으나, 웹 프로그래밍의 기초가 되는 서버 측 동적 스크립트 언어와 관계형 데이터베이스(RDBMS) 간의 연동 원리를 명확히 이해하는 것은 여전히 고성능 웹 시스템을 설계하는 데 있어 필수적인 기본 토대이다[4,5]. 또한, 서버 측 개발 언어인 JSP(Java Server Page)는 플랫폼 독립적인 Java 기술을 기반으로 동적 웹 페이지 생성을 용이하게 하고, MySQL은 우수한 처리 성능과 높은 신뢰성을 바탕으로 중소 규모부터 대규모 전자상거래 시스템에 이르기까지 광범위하게 활용되고 있다[6,7].

따라서 본 논문에서는 JSP와 MySQL을 이용한 온라인 쇼핑몰 사이트 개발 과정 전반에 걸친 주요 모듈별 프로세스 흐름 및 연관관계, 데이터베이스 모델링, 그리고 주요 모듈별 핵심 알고리즘들을 종합적으로 심층 분석 및 제시하고자 한다. 또한, '즉시 구매' 처리를 위해 단기 생명주기 기반의 '임시 장바구니(tempcart) 개념'을 도입하여, 기존 장바구니 구현 시에 사용되었던 소스코드들을 상당부분 그대로 재활용하면서도 웹 애플리케이션 서버(Web Application Server, WAS)의 힙(Heap) 메모리 부하를 원천 차단할 수 있는 효율적인 설계 방안을 제안하였다. 나아가 본 논문의 연구 결과는 신규 웹 개발 인력 및 관련 분야 학습자들에게 체계적인 온라인 쇼핑몰 사이트 개발 가이드라인을 제공하는데 기여할 수 있을 것으로 기대된다.

2. 구현 환경

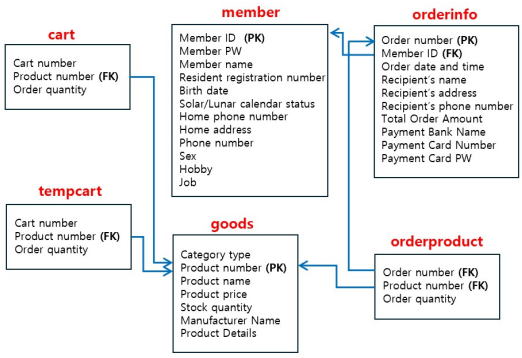
서버 측 애플리케이션 개발 언어로는 Java 기반의 대표적인 동적 웹 스크립트 기술인 JSP를 이용하였다. 클라이언트 측 프론트엔드 사용자 인터페이스는 HTML5, CSS3 및 JavaScript를 조합하여 적용하였다. 이를 통해 사용자 중심의 반응형 웹 컴포넌트를 구성하고, 사용자 입력값의 1차 검증을 수행하여 불필요한 서버 요청 트래픽을 차단하도록 설계하였다. 본 사이트가 작동하는 서버 측 운영체제 환경으로는 Windows 11을 사용하였으며, Java 애플리케이션의 개발 및 실행 컴파일러로는 JDK(Java Development Kit) 8.0을 이용하였다. JSP 페이지의 변환 및 서블릿 실행을 담당하는 WAS로는 Apache Tomcat 9.0 컨테이너를 탑재하여 동적 세션 관리와 HTTP 요청처리의 안정성을 확보하였다[8]. 백엔드 데이터베이스 관리시스템(DBMS)으로는 MySQL 8.0을 채택하였다. 서버 애플리케이션과 원격지 MySQL 간의 효율적인 데이터 상호작용 및 통신 인프라를 형성하기 위해 전용 JDBC(Java DataBase Connectivity) 드라이버인 Connector/J 8.0 라이브러리를 연동하여 최적의 데이터 입출력 환경을 구현하였다[9].

3. 시스템 설계 및 구현

3.1 데이터베이스 모델링

온라인 쇼핑몰 사이트의 성공적인 구현 및 유연한 데이터 관리를 위해서는 웹 클라이언트와 원격지 DB 서버 간의 일관성 있는 데이터 연동 체계 구축이 필수적이다. 이를 실현하기 위한 핵심 전제 조건으로서 체계적인 데이터베이스 모델링 과정이 요구된다. 본 연구에서 정의한 온라인 쇼핑몰 사이트의 논리적 데이터베이스 구조는 [Fig. 1]과 같다. [Fig. 1]의 각 엔티티 명칭에 표기된 영문 명칭은 물리적 모델링 단계에서 생성된 데이터베이스의 실제 테이블명에 대응되며, 'PK' 및 'FK'는 각각 엔티티의 무결성을 보장하기 위한 기본 키(Primary Key)와 타 엔티티 참조를 위한 외래 키(Foreign Key)를 의미한다. [Fig. 1]은 회원 정보를 관리하는 member 테이블, 판매 상품 명세를 관리하는 goods 테이블, 장바구니 및 임시 장바구니를 담당하는 cart와 tempcart 테이블, 그리고 주문 정보 및 주문 상세 항목을 관

리하는 orderinfo와 orderproduct 테이블들 간의 유기적인 관계망을 형성한다. 또한, 이러한 데이터베이스 모델링은 표준 웹 브라우저 기반의 입력 환경을 충실히 지원하는 동시에, 향후 오프라인 스마트 리테일 매장의 무인 카트나 IoT 센서 등 다양한 스마트 디바이스 환경에서 유입되는 실시간 상품 및 주문 데이터 인터페이스까지 유연하게 수용할 수 있는 구조적 확장성을 고려하여 설계되었다[2,10,11].



[Fig. 1] Logical Database Structure

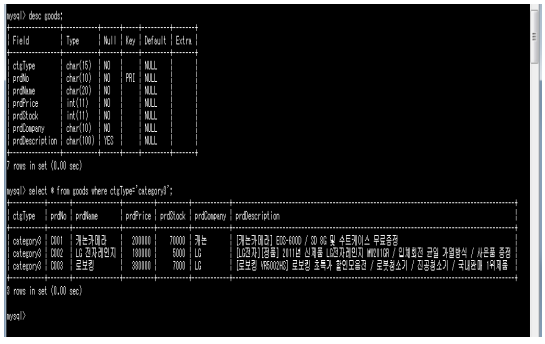
3.2 상품 관리

Java/JSP 기술 스택을 활용하여 웹 클라이언트와 원격지 DB 서버 간의 정합성 있는 데이터 연동을 구현하기 위해서는, 데이터베이스의 기본 조작 기능인 CRUD 연동 프로세스가 체계적으로 정의되어야 한다. 본 연구에서는 JDBC 연동 표준 프로그래밍 절차에 입각하여 상품 관리 모듈의 제반 기능들이 JSP로 구현되었다. [Fig. 2]는 웹 클라이언트 환경에서 사용자가 카테고리별 상품 조회(예: '디카/주방가전')를 요청했을 때, 서버가 동적으로 생성하여 웹 브라우저에 출력한 카테고리별 상품 조회된 결과 화면의 예시이다.



[Fig. 2] Results Screen Retrieved from the Web via Product View by Category

[Fig. 3]은 원격지 서버상의 MySQL 데이터베이스에 구축된 상품 테이블(goods)의 물리적 구조 명세 및 실제 SQL 문을 통한 카테고리별 상품 조회(SELECT * FROM goods WHERE ctgType='category3';)가 실행된 예시 결과를 나타낸다. [Fig. 2]와 [Fig. 3]의 비교 분석을 통해, 원격지 MySQL 데이터베이스에 저장된 상품 레코드의 필드 데이터들이 웹 클라이언트 인터페이스상에 왜곡 없이 완벽하게 동일한 데이터 정합성을 유지하며 동적으로 표현되고 있음을 확인할 수 있다. 한편, 지면 관계상 상세한 실행 화면 제시는 생략하나, 상품 정보의 신규 등록(INSERT), 기존 상품 정보의 수정(UPDATE), 그리고 상품 정보의 삭제(DELETE) 연동 프로세스 역시 JDBC 표준 처리 알고리즘을 엄격히 준수하여 구현되었다.

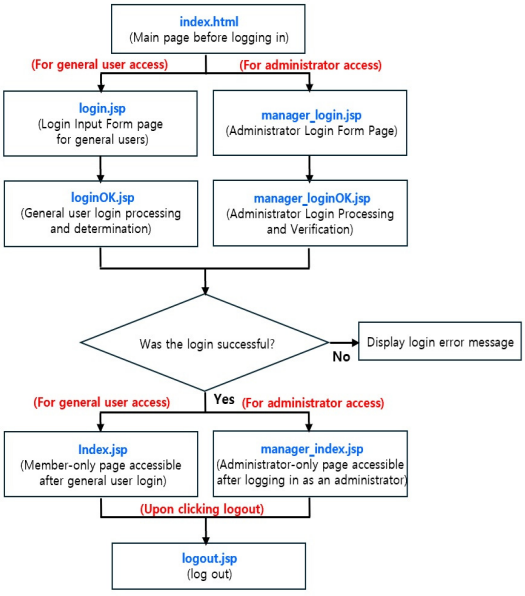


[Fig. 3] Physical Structure of the 'goods' Table on a Remote MySQL Instance

3.3 회원 관리 및 사용자 인증

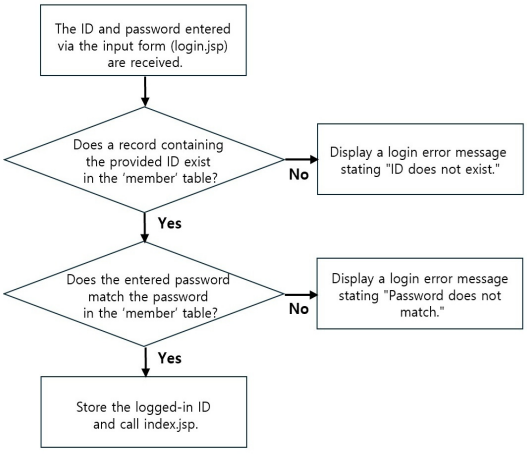
온라인 쇼핑몰 사이트에서 회원 관리 및 사용자 인증 모듈은 인가된 사용자와 관리자의 접근 권한을 명확히 분리하고 개인화된 서비스를 제공하기 위한 핵심적인 보안 기능적 기본 요소이다. [Fig. 4]는 사용자 인증 및 로그인 프로세스를 처리하기 위한 관련된 소스코드들 간의 유기적 흐름과 세션 제어 흐름도를 나타낸다. [Fig. 4]와 같이, 본 사이트는 메인 페이지(index.html) 접근 시 일반 사용자 접속 경로와 관리자 접속 경로를 이원화하여 설계하였다. 일반 사용자가 로그인 폼(login.jsp)을 통해 접속을 시도하면 사용자 인증처리 모듈(loginOK.jsp)로 전달되며, 관리자는 전용 로그인 페이지(manager_login.jsp)를 거쳐 관리자 인증 처리 모듈(manager_loginOK.jsp)에서 검증 절차를 거치게 된다. 인증 결과의 성공 여부에 따라 접근 허용 페이지(index.jsp 또는 manager_index.jsp)로 리다이렉트(redirect) 되거나 에러 메시지를 출력하며, 로그아웃 요청 시 세션을 완전히 파기하는 세션 관리 모듈(logout.jsp)을 호출하도록 구성하였다.

www.kci.go.kr



[Fig. 4] Process Flow Diagram
(User Authentication and Login)

사용자 인증 처리의 중추 역할을 담당하는 loginOK.jsp의 세부 알고리즘은 [Fig. 5]와 같다. 정상적인 사용자 로그인 및 세션 생성이 이루어지기 위해서는 클라이언트가 입력 폼에서 전송한 사용자 아이디(ID) 및 비밀번호(PW) 데이터가 원격지 MySQL 데이터베이스 내 회원 관리 테이블(member)의 레코드와 엄격히 대조 검증되어야 한다. 인증 메커니즘은 [Fig. 5]에서 보여지는 순차적 논리 검증 단계를 거쳐 수행된다.



[Fig. 5] Key Algorithms Related to User Authentication (loginOK.jsp)

한편, 지면 관계상 상세한 실행 화면 제시 및 개별 코드는 생략하나, 신규 회원가입 모듈을 통해 데이터베이스에 성공적으로 저장된 회원 정보는 회원 ID, 비밀번호, 성명, 생년월일, 연락처, 주소 등의 속성 필드들로 구성된다. 입력 폼을 통해 전달된 계정 정보가 회원 관리 테이블상의 필드 데이터들과 정확히 일치하는 경우에만 인증이 성립되며, 신규 회원 등록, 회원 정보 조회 및 수정, 회원 탈퇴 역시 동일한 데이터 무결성 기준에 따라 원격지 DB와 연동됨을 확인하였다.

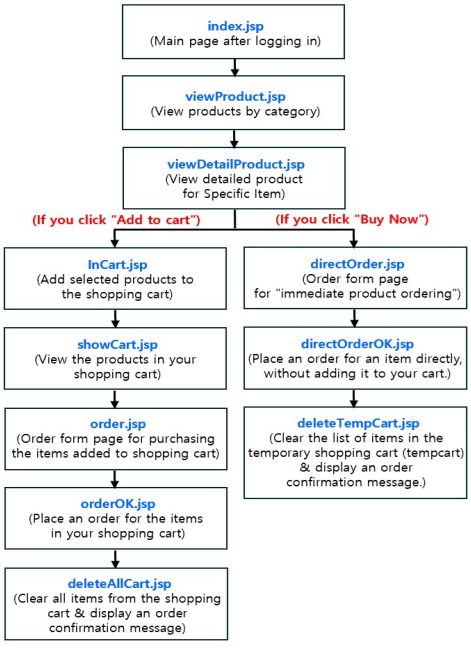
3.4 장바구니 및 주문 관리

온라인 쇼핑몰 사이트 구축에 있어 장바구니 및 주문 관리 모듈은 웹 클라이언트에서 선택된 상품 정보가 원격지 데이터베이스에 원활히 반영되고 결제 및 주문 처리까지 안정적으로 이어지도록 하는 핵심 비즈니스 로직이다. [Fig. 6]은 웹 클라이언트 브라우저상에 출력된 상품 상세 페이지(viewDetailProduct.jsp)의 예시이다. 상세페이지는 URL 전달인자로 넘겨받은 상품의 기본키인 상품번호(prdNo)를 기반으로 해당 상품의 상세 정보 및 이미지를 동적으로 출력하며, 사용자가 선택한 주문 수량을 바탕으로 '장바구니 담기'와 '즉시 구매하기'의 이원화된 구매 경로를 제공한다. 사용자의 주문 버튼 선택에 따른 전체적인 프로세스 흐름은 [Fig. 7]와 같다. [Fig. 8]은 '장바구니 담기' 기능(inCart.jsp)의 세부처리 알고리즘을 나타낸다. 본 모듈 실행 시 가장 먼저 세션 상태를 확인하여 미인증된 사용자의 경우 로그인 요청 안내 메시지를 출력하도록 제어한다. 인증된 사용자 요청에 대해서는 원격지 데이터베이스의 장바구니 테이블(cart)을 조회한 후 [Fig. 8]에서 보여지는 단계적 제어 논리 알고리즘을 수행한다. [Fig. 9]는 '장바구니 보기' 기능(showCart.jsp)의 내역 추출 알고리즘을 나타낸다. cart 테이블 내 레코드가 존재하지 않을 경우 "장바구니가 비었습니다"라는 안내 메시지를 반환한다. 레코드가 존재할 경우, cart 테이블에 존재하는 레코드 개수만큼 반복문을 수행하면서 cart 테이블에서 상품번호(prdNo)와 주문수량(ctQty)을 추출하고, 상품 테이블(goods)과의 연동을 통해 상품명(prdName)과 상품단가(prdPrice)를 추출한다. 서버 프로그램에서는 이를 바탕으로 개별 상품별 주문액(amount = prdPrice × ctQty)과 전체 주문총액(total)을 연산하여 웹 브라우저 화면에 동적으로 출력한다. 최종 주문 신청 단계인 orderOK.jsp 모듈의 실행 알고리즘은 [Fig. 10]과 같다. 사용자가 주문서 입력 페이지(order.jsp)에서 주문을 확정하면, 주문 정보 테이블(orderinfo)로부터 현재 부여된 마지막 주문번호의

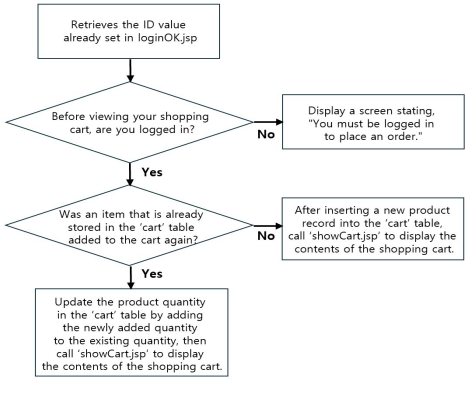
www.kci.go.kr



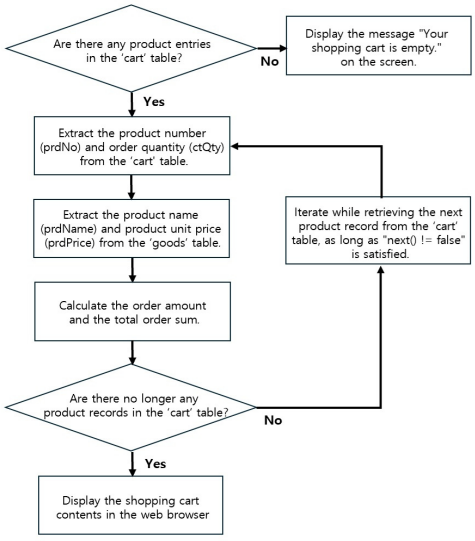
[Fig. 6] Example of a Product Detail Page



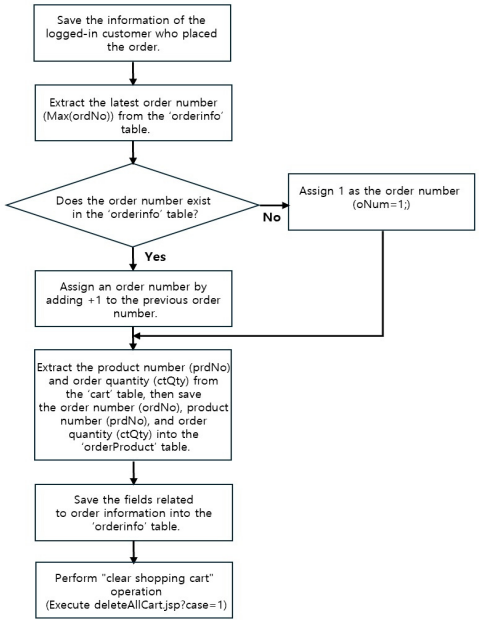
[Fig. 7] Process Flow Diagram
(from Product View to Placing an Order)



[Fig. 8] Key Algorithms Related to the 'Add to Cart' (inCart.jsp)



[Fig. 9] Key Algorithms Related to the 'Shopping Cart View' (showCart.jsp)



[Fig. 10] Key Algorithms Related to Placing an Order (orderOK.jsp)

최대값(Max(ordNo))을 추출한다. 기존 주문 내역이 없는 초기 상태일 경우 주문번호를 1로 할당(oNum = 1)하고, 내역이 존재할 경우 기존 최대값에 1을 더하여 새로운 주문번호를 할당(oNum = rs.getInt(1) + 1)함으로써 주문 식별자의 유일성을 확보하였다. 이후 cart 테이블로부터 추

www.kci.go.kr

출한 상품 정보 및 새로 할당된 주문번호를 주문 상품 상세 테이블(orderproduct)에 개별 레코드로 저장하며, 주문자 정보 및 결제/수령인 관련 데이터는 orderinfo 테이블에 통합 저장함으로써 데이터 정규화를 준수하였다. 주문 완료 후에는 장바구니를 비우는 모듈(deleteAllCart.jsp)을 호출하게 된다. 본 연구에서는 사용자가 단순 변심 등으로 장바구니를 명시적으로 비우는 상황과, 정상적인 주문 완료 결과로서 장바구니가 비워지는 상황을 논리적으로 구분하기 위해 case 파라미터를 전달인자로 활용하였다. 즉, deleteAllCart.jsp?case=1 과 같이 파라미터를 다르게 넘겨줌으로써 각 상황에 상응하는 안내 메시지 출력 및 후속 페이지 리다이렉트가 적절히 이루어지도록 처리하였다.

한편, 단일 상품을 즉시 결제하는 '즉시 구매하기' 기능의 구현에 있어서 기존 장바구니 주문 알고리즘과의 충돌을 방지하고 소스코드 재활용성(Reusability)을 극대화하기 위해 다음과 같은 구조적 설계를 적용하였다. 일반적인 장바구니 주문 경로가 [Fig. 7]의 좌측과 같이 상품의 장바구니 적재(inCart.jsp), 장바구니 내역 조회(showCart.jsp), 그리고 주문서 작성(order.jsp)의 다단계 소스코드 및 페이지를 순차적으로 거치는 것과 달리, 본 연구에서 제안하는 즉시 구매 모듈(directOrder.jsp)은 위 세 가지 기존 모듈이 수행하던 역할과 핵심 로직을 단일 파일 내에서 통합 담당하도록 설계되었다. 특히, directOrder.jsp 내부의 데이터 추출, 가격 연산 및 입력 폼 구성 로직은 기존 장바구니 소스코드들을 상당 부분 그대로 재사용하여 구현되었다. 이를 통해 상품 정보 조회 및 가격 연산 로직의 중복 코딩을 방지하고 유지보수성을 대폭 향상시켰으며, cart 테이블과 동일한 필드 구조를 갖는 '임시 장바구니 테이블(tempcart)'을 백엔드에서만 독립적으로 운용함으로써 기존 시스템과의 간섭을 원천 차단하였다. 이와 함께, 백엔드에서 운용되는 tempcart 테이블은 즉시 구매 시점에 선택된 단일 레코드만을 임시적으로 적재한 후, 주문 완료 즉시 하위 삭제 모듈(deleteTempCart.jsp)을 통해 레코드가 곧바로 영구 삭제되도록 구현되었다. 이러한 데이터 생명주기의 단기성(Transient)은 '세션(Session) 기반의 전통적인 즉시 구매 구현 방식'이 지닌 한계를 효과적으로 극복한다. 즉, 수많은 다중 사용자들의 동시 '즉시 구매' 시 각 사용자별 세션 객체(HttpSession) 및 주문 데이터가 WAS의 자바 가상 머신(Java Virtual Machine, JVM)의 힙(Heap) 메모리상에 동시에 대량 적재되어 발생될 수 있는 메모리 과부하, 가비지 컬렉션(GC) 오버헤드, 그리고 세션 유실에 따른 데이터 정합성 훼손 위험을 원천 차단하고 WAS를 경량화된 구조로 유지하도록 돕는다.

4. 성능 평가 및 검증

본 장에서는 '세션 기반의 전통적인 즉시 구매 구현 방식'과의 상대 비교 실험은 별도로 수행하지 않았으나, 제안하는 tempcart 아키텍처가 적용된 통합 시스템의 절대적 성능과 유효성을 엄격히 검증해 보려는 데 주력하였다. 본 연구에서 제안한 'tempcart 기반의 즉시 구매 구현 방식'의 다중 트래픽 환경 내 처리 성능과 데이터 정합성 보장 메커니즘을 검증하기 위한 성능 테스트 도구로는 부하 생성 표준 도구인 Apache JMeter[12]를 활용하였으며, 소규모 접속부터 대규모 동시 접속 상황에 이르는 총 5가지 시나리오를 구성하였으며, 각 시나리오별로 4회씩 반복 측정된 결과의 산술 평균 및 대표값들을 〈Table 1〉에 정리하였다.

4.1 실험 환경 및 제어 변수 설정

성능 평가 실험은 웹 서버(Apache Tomcat)와 데이터베이스(MySQL)가 동일한 시스템 내에서 프로세스 간 통신을 수행하는 로컬 환경에서 진행되었다. 실험 과정에서 발생할 수 있는 운영체제 및 네트워크의 일시적 측정 오차를 최소화하고 데이터의 객관성을 확보하기 위해, 각 시나리오별로 동일한 진입시간(Ramp-up Period, 10초)과 반복횟수(Loop Count, 5회) 조건 하에서, 각 시나리오별로 4회씩 반복 측정을 수행하였다. 여기서 진입시간(10초)은

〈Table 1〉 Performance Analysis Results

Test Scenario	Threads (Number of Users)	Loop Count	Number of Order Transactions	Average Response Time	Min	Max	Standard Deviation	Error Rate	Throughput (TPS)
1	50	5	250	1.4ms	0ms	27ms	1.79	0.0%	25.32/sec
2	100	5	500	1.2ms	0ms	35ms	1.61	0.0%	50.36/sec
3	200	5	1000	1.2ms	0ms	52ms	1.53	0.0%	100.14/sec
4	300	5	1500	0.9ms	0ms	59ms	1.69	0.0%	151.66/sec
5	500	5	2500	1.2ms	0ms	39ms	1.89	0.0%	249.56/sec

www.kci.go.kr

설정된 가상 동시 사용자(Threads)가 점진적으로 서버에 유입되도록 제어하여 실제 웹 서비스의 트래픽 진입 양상을 모사하기 위함이며, 반복횟수(5회)는 각 사용자가 주문 트랜잭션을 연속 수행하게 함으로써 최대 2,500건에 이르는 누적 부하 상황에서도 시스템이 성능 저하 없이 일관된 안정성을 유지하는지 검증하기 위해 통제 변수로 고정하였다.

4.2 동시성 제어 및 트랜잭션 무결성 검증

다중 사용자의 동시 주문 요청 시 발생할 수 있는 식별자 중복 및 데이터 불일치 문제를 방지하기 위해 다음과 같은 안정성 메커니즘을 적용하였다.첫째, 주문번호(ordNo)를 데이터베이스의 기본 키로 설정하여 동시 입력 상황에서 물리적인 식별자 유일성(Uniqueness)을 엄격히 보장하고 중복 입력을 원천 차단하였다. 둘째, 경량화된 순수 JSP 및 JDBC 기반 아키텍처 환경의 특성을 고려하여, 주문 마스터 정보와 주문 상세 정보의 저장을 하나의 원자적 작업 단위(Atomic Transaction)로 묶었다. 이를 위해 JDBC 수동 트랜잭션 제어 방식(setAutoCommit(false))을 적용하였으며, 처리 과정 중 예외가 발생할 경우 즉시 전체 작업을 취소하는 rollback()을 수행하도록 설계하여 데이터의 부분 저장으로 인한 정합성 훼손을 방지하였다.

4.3 성능 평가 결과 및 지표 해석

5개의 각 시나리오별로 4회씩 반복 측정된 결과의 산술 평균 및 대표값을 정리한 최종 성능 평가 결과는 〈Table 1〉과 같다. 단, 시스템의 최악 지연 시간(Worst-Case)을 통한 안정성 검증을 위해 최소 응답 시간(Min)은 4회 측정 중 최저값을, 최대 응답 시간(Max)은 4회 측정 중 관측된 최고값을 기재하였다. 성능 측정 결과의 분석 내용은 다음과 같다.

(1) 에러율 및 시스템 무결성: 50명에서 최대 500명의 동시 접속(총 2,500건 트랜잭션)이 발생하는 모든 시나리오에서 에러율 0.00%를 기록하였다. 이는 앞서 설계한 기본 키 유일성 제어 및 JDBC 트랜잭션 롤백 메커니즘이 대규모 부하 상황에서도 충돌이나 실패 없이 무결하게 작동함을 증명한다.

(2) 응답 속도 및 초고속 처리 능력: 5가지의 모든 시나리오들에 걸쳐 평균 응답 시간(Average)이 0.9ms~1.4ms 내외로 수렴하였으며, 최소 응답 시간(Min)은 0ms로 나타났다. 이는 로컬 환경의 네트워크 지연 부재와 측정 도구(JMeter)의 밀리초 반올림 단위 미만으로 단축될 만큼 초고속 처리가 이루어짐을 보여준다.

(3) 최대 지연 시간(Max)의 안정적 통제: 4회 반복 측정 중 가장 악조건에서 관측된 최대 응답 시간(Max)은 200~300명 구간에서 일시적 프로세스 스케줄링 간섭 등의 영향으로 추정되는 이유로 인해 각각 52ms 및 59ms를 기록하였으나, 500명 최대 부하 시에도 39ms 이하로 안정적으로 제어되었다. 이는 사용자 인지 한계 시간(약 100ms)을 대폭 하회하는 수치로서, 최악의 상황에서도 병목 현상 없이 원활히 응답함을 의미한다.

(4) 선형적 처리량(Throughput) 확장성: 동시 사용자 수가 50명에서 500명으로 10배 증가함에 따라 초당 처리량(Throughput) 또한 25.32/sec에서 249.56/sec로 비례하여 완벽한 선형 우상향 추세를 나타내었다. 이는 제안하는 임시 장바구니 아키텍처가 트래픽 급증 시에도 높은 수평적 확장성과 자원 소화 능력을 제공함을 입증한다.

5. 결론

본 논문에서는 최근 스마트 커머스 및 비대면 전자상거래 환경의 급속한 확대에 발맞추어, 온라인 쇼핑몰 사이트를 보다 체계적이고 효율적으로 구축하기 위한 전반적인 모듈별 프로세스 흐름과 데이터베이스 모델링, 그리고 주요 모듈별 핵심 알고리즘들을 제시 및 분석하였다. 특히, 단일 상품에 대한 '즉시 구매' 처리를 위해, 단기 생명주기(Transient) 기반의 임시 장바구니(tempcart) 아키텍처를 새롭게 도입·정립하여, 기존 장바구니 소스코드를 상당 부분 그대로 재활용하는 코드 재사용성 극대화와 더불어, 수많은 다중 사용자 접속 시에도 WAS 힙(Heap) 메모리 과부하 및 세션 유실 위험을 원천 차단하는 구조적 안정성을 동시에 달성하였다. 한편, 본 연구에서 정립한 프로세스 모델과 데이터베이스 아키텍처는 특정 서버 스크립트 언어나 DBMS에 종속되지 않는 보편적 확장성을 지닌다. 즉, 세션 기반의 사용자 인증, 정규화된 RDBMS 엔티티 연동 구조, 그리고 tempcart를 활용한 단기 상태 관리 및 모듈 통합 구조는 동일한 관계형 데이터베이스 모델을 지원하는 ASP.NET, PHP 등 타 웹 개발 환경과 Oracle, MS-SQL 등의 상용 DBMS 환경에서도 표준 데이터베이스 커넥션 규격(JDBC, ODBC 등)을 통해 동일한 논리로 용이하게 확장 및 적용될 수 있다.

www.kci.go.kr

그러나 본 연구는 단일 서버 및 기초 JDBC Connection 연동 방식을 중심으로 구현되어, 대규모 동시 접속자가 발생하는 환경에서의 성능 최적화(Connection Pool 적용 등) 및 개인정보 보호를 위한 데이터 암호화 등 최신 웹 보안 프레임워크 적용 측면에서 다소 한계점을 지니고 있다[13,14,15]. 특히 실제 상용 웹 서비스 구축 시에는 평문 인증 방식에서 탈피하여 단방향 해시 알고리즘(SHA-256/BCrypt) 기반의 비밀번호 암호화 저장·검증과, 세션 위·변조 방지 및 사용자별 장바구니 데이터에 대한 수평적 접근 통제 메커니즘이 필수적으로 결합되어야 한다. 향후 연구에서는 이러한 기술적 부재와 보안적 한계를 극복하기 위해 커넥션 풀 기법 도입과 함께 엄격한 웹 보안 프레임워크를 결합하여 시스템의 안정성과 보안성을 한층 고도화하고자 한다.

REFERENCES

[1] J.H.Paek, J.S.Park and M.W.Hong, “Building an E-commerce Store Using IBM Net.Commerce”, Jungiksa, 2000

[2] J.S.Lee and M.S.Choi, “Implementation of a Real-time Inventory Management and Smart Shopping Mall System Based on IoT Sensor Data,” Journal of IoT Convergence, Vol.8, No.2, pp.15-22, 2022.

[3] D.G.Kim and J.W.Lee, “Design and Implementation of an Open Source-based E-Commerce Platform for Small Business Owners,” Journal of the Korea Society of Computer and Information, Vol.26, No.4, pp.45-53, 2021.

[4] H.C.Park and S.W.Lee, “A Study on Connection Pool Performance Optimization for Web Applications Handling High-Volume Traffic,” Journal of Korean Institute of Information Technology, Vol.18, No.8, pp.115-123, 2020.

[5] P.Gupta and R.Singh, "Security Challenges and Countermeasures in Modern Web-based E-Commerce Systems," Journal of King Saud University - Computer and Information Sciences, Vol.34, No.6, pp.3102-3115, 2022.

[6] Oracle Corporation, "Java SE Downloads," Oracle Technology Network. [Online]. Available: https://www.oracle.com/kr/java/technologies/downloads/.

[7] Oracle Corporation, "MySQL Community Server," MySQL Developer Zone. [Online]. Available: https://dev.mysql.com/downloads/.

[8] Apache Software Foundation, "Apache Tomcat," [Online]. Available: https://tomcat.apache.org.

[9] Oracle Corporation, "MySQL Connector/J," MySQL Developer Zone. [Online]. Available: https://dev.mysql.com/downloads/connector/j/.

[10] Apache Software Foundation, "Apache JMeter," [Online]. Available: https://jmeter.apache.org.

[11] Y.Zhang and X.Chen, "Performance Optimization of Relational Database in E-Commerce Platforms," IEEE Access, Vol.9, pp.45120-45129, 2021.

[12] S.H.Choi and H.B.Kim, “Analysis of OWASP Top 10 Vulnerabilities in Web Applications and Design of a Security Patch Framework,” Journal of the Korea Institute of Information Security & Cryptology, Vol.30, No.5, pp.889-897, 2020.

[13] T.H.Im and M.W.Park, “Implementation of a Secure Session Management and Authentication Framework to Prevent Session Hijacking in Web Applications,” Journal of Convergence Security, Vol.22, No.3, pp.77-85, 2022.

[14] J.W.Jeong and S.H.Park, “User Location Awareness and Personalized Product Recommendation System for Beacon-based O2O E-commerce Services,” Journal of the Korea Internet of Things Society, Vol.7, No.3, pp.33-40, 2021.

[15] S.H.Yoon, “A Study on Asynchronous Transaction Processing and Database Performance Improvement Methods for E-commerce Systems,” Journal of the Korea Contents Association, Vol.21, No.6, pp.12-22, 2021.

공 기 식(Ki-Sik Kong) [정회원]



- 1999년 2월 : 고려대학교 컴퓨터학과 (이학학사)
- 2001년 8월 : 고려대학교 컴퓨터학과 (이학석사)
- 2005년 8월 : 고려대학교 컴퓨터학과 (이학박사)
- 2005년 9월 ~ 2009년 2월 : 고려대학교 정보통신대학 연구교수
- 2009년 3월 ~ 현재: 남서울대학교 멀티미디어학과 교수

〈관심분야〉

모바일네트워크, 정보통신, 사물인터넷, 웹 애플리케이션

www.kci.go.kr