

스프링 프레임워크 환경에서 스프링 데이터 JPA기반의 엔터프라이즈 시스템 플랫폼의 설계

유정상¹, 이명호^{2*}

¹가천대학교 산업경영공학과 교수, ²세명대학교 전자상거래학과 교수

Design of Enterprise System Platform based on Spring Data JPA in Spring Framework Environment

Jung-Sang Yoo¹, Myeong-Ho Lee^{2*}

¹Professor, Department of Industrial Engineering, Gachon University

²Professor, Department of eCommerce, Semyung University

요약 엔터프라이즈 환경의 표준화 경쟁은 백엔드의 데이터 티어로 시작하여 대표적인 엔터프라이즈 미들 티어가 스프링 프레임워크로 받아들여짐으로써 표준화로 안정화되고 있는 실정이다. 또한 점차 빠른 주기로 새로운 디바이스의 출현으로 웹과 모바일 서비스에 대한 호환성 확보가 웹 서비스 기업들의 중요한 경쟁력이 되고 있다. 그러나 국내 기업들은 이러한 정보화 시대의 격변한 환경 변화에 적절한 역량있는 기술 인력을 확보하지 못하고 있으며, 교육중심 대학들의 교육과정에서도 새로운 역량중심의 교육과정의 요구를 반영하지 못하고 있는 실정이다. 따라서 본 연구에서는 이러한 엔터프라이즈 시스템 플랫폼 환경에서 필요한 역량중심의 기술을 습득과 교육과정을 개발하기 위하여 스프링 프레임워크 환경에서 스프링 데이터 JPA를 활용한 시스템을 분석 및 설계 단계별로 문서화 작성을 통하여 구현하였다. 향후 엔터프라이즈 환경에서의 바로 적용할 수 있는 풀 스택 역량중심의 교육과정 및 캡스톤 디자인 교육과정의 참조 모델을 제공하고자 한다.

주제어 : 엔터프라이즈, 표준화, 웹, 모바일, 스프링 프레임워크, 스프링 데이터 JPA, 풀 스택

Abstract The competition for standardization in the enterprise environment starts with the data tier of the back-end and the standard enterprise middle tier is being stabilized by standardization as it is accepted as the Spring Framework. In addition, with the advent of new devices in an increasingly rapid cycle, securing compatibility with web and mobile services has become an important competitive advantage for web service companies. However, companies are unable to secure competent technical personnel appropriate for the rapidly changing environment of the information generation, and the curriculum of educational universities does not reflect the demand of new competency-oriented curriculum. Therefore, in this study, in order to acquire competency-oriented skills required for such an enterprise system platform environment and to develop a curriculum, the system using Spring Data JPA in the Spring Framework environment was implemented through documenting for each analysis and design step. It aims to provide a reference model for the full stack competency-based curriculum and capstone design curriculum that can be applied immediately in the enterprise environment.

Key Words : Enterprise, Standardization, Web, Mobile, Spring Framework, Spring Data JPA, Full Stack

*Corresponding Author : Myeong-Ho Lee(mhlee@semyung.ac.kr)

1. 서론

정보화 시대의 컴퓨팅 흐름은 1970년대까지 메인 프레임 시대에서 1990년대 말까지 클라이언트-서버 컴퓨팅 시대를 맞이하게 되었다. 그러나 1989년 CERN의 팀 버너스리의 제안으로 시작된 인터넷의 WWW(World Wide Web) 관련 기술들은 마이크로 컴퓨터 성능의 비약적인 발전과 함께 새로운 정보화 패러다임을 완전히 바꾸어놓게 되었다[1].

1990년 후반부터는 인터넷의 확산으로 웹 브라우저를 사용한 전자상거래의 요구가 생기면서 새로운 정보화 변화의 물결이 나타나게 되었다. 그것은 바로 웹 애플리케이션 서버(WAS)였으며 웹 애플리케이션과 서버 환경을 만들어서 동작되는 미들웨어 소프트웨어 프레임워크이다[2]. 이때부터 정보화 시대의 새로운 미들웨어에 대한 표준화 선점은 서비스 기반의 엔터프라이즈 비즈니스 생태계를 완전히 기업들의 생존 경쟁으로 나타나게 되었다. 2007년도의 애플의 새로운 디바이스인 아이폰(iPhone)의 등장으로 정보화 시대는 웹의 시대에서 모바일 시대를 위한 전환점의 서막을 알리게 되었다[3]. 기업들의 비즈니스 SOA (Service-Oriented Architecture) 환경도 웹 중심에서 모바일 우선 전략 중심으로 이동하는 계기가 되었다[4,5]. 이러한 현상은 정보 보안으로 인한 웹서비스(Web Services)의 표준화 실패에 따라 2010년 이후로 새로운 클라우드 컴퓨팅 플랫폼의 출현으로 나타나게 되었다[6].

따라서 2000년 이후로 컴퓨터 환경의 플랫폼도 분산 컴퓨팅 환경을 위한 N-tier 자바 플랫폼인 Java EE가 엔터프라이즈 시스템 플랫폼의 중심으로 자리 잡게 되었다[7]. 그러나 Java EE(J2EE)의 복잡한 사양서는 점차 서비스 기반 비즈니스에서 외면 받기 시작하면서, 2002년 로드 존슨(Rod Johnson)에 의해 제안된 스프링 프레임워크(Spring Framework)는 2004년 1.0 버전 이후 Java EE의 표준 환경 오픈소스 애플리케이션 프레임워크로 발전하게 되었다[8]. 이와 같은 환경의 변화에 따라 행정안전부에서는 2010년 6월 전자정부서비스 호환성 준수지침(제 2010-40호)의 고시하였으며[9], 2011년 9월에는 NIA(한국정보화진흥원)를 통하여 전자정부 표준프레임워크 개발환경 2.0.0을 시작으로 2019년 현재 개발환경 3.8.0이 스프링 프레임워크를 기반으로 발표하였다[10]. 또한 데이터 티어(Data Tier)의 표준화

이후 미들 티어(Middle Tier)인 WAS의 표준화 선점 경쟁은 2010년도까지 많은 사양서(Specification)들이 발표되면서 엔터프라이즈 환경은 기업들의 경쟁력 제고에 많은 영향을 주었다. 현재 국내 기업들은 이러한 정보화 시대의 격변한 환경 변화에 따라 역량있는 인력 수급이 필요한 시점이며, 교육중심 대학들의 교육과정에서도 많은 문제점과 새로운 역량 중심의 교육과정의 요구가 도출되고 있는 실정이다 [11,12].

따라서 본 연구에서는 이러한 엔터프라이즈 시스템 플랫폼 환경에서 필요한 역량중심의 기술을 습득하기 위하여 스프링 프레임워크 5 환경에서 스프링 데이터 JPA(Java Persistence API)를 활용한 시스템을 분석 및 설계한 후 구현함으로써, 향후 엔터프라이즈 환경에서의 바로 적용할 수 있는 풀 스택 역량중심의 교육 과정 및 캡스톤 디자인 교육과정의 참조 모델을 제공하고자 한다.

2. 엔터프라이즈 플랫폼에 대한 고찰

2.1 스프링 프레임워크 5

2004년 3월에 첫 번째 스프링 프레임워크 1.0이 릴리즈 된 이후, 2009년 12월 Java5 및 REST API를 지원하는 스프링 프레임워크 3.0을 거쳐, 2013년 12월에는 JPA 2.0과 Servlet 3.0 등을 지원하는 스프링 프레임워크 4.0이 발표되었다. 현재는 2017년 9월 JavaSE 8의 특징인 강화된 랬다식 표현, 스트림과 병렬 스트림, 디폴트 메서드를 지원하는 스프링 프레임워크 5가 발표되었다[13-15].

2.2 스프링 MVC의 구조

스프링 MVC는 먼저 웹 브라우저의 요청이 들어오면 처리할 컨트롤러 빈 객체를 디스패처 서블릿에 전달한다. 디스패처 서블릿은 핸들러 매핑이 찾아준 컨트롤러 객체를 핸들러 어댑터 빈에게 요청 처리를 위임한다. 핸들러 어댑터는 요청을 처리한 후 결과를 디스패처 서블릿에 리턴한다. 이때 핸들러 어댑터는 처리 결과를 모델엔뷰라는 객체로 변환해서 디스패처 서블릿에 리턴한다. 핸들러 어댑터로부터 요청 처리 결과를 모델엔뷰로 받으면 디스패처 서블릿은 결과를 보여줄 뷰를 찾기 위해 뷰 리졸버 빈 객체를 사용한다. 모델엔

뷰는 컨트롤러가 리턴한 뷰 이름을 담고 있다. 뷰 리졸버는 이 뷰 이름에 해당하는 뷰 객체를 찾거나 또는 생성해서 리턴해 준다. 응답을 생성하기 위해 JSP를 사용하는 뷰 리졸버는 매번 새로운 뷰 객체를 생성해서 디스패처 서블릿에 리턴한다. JSP를 사용하면 뷰 객체는 JSP를 실행하여 브라우저에 전송할 응답 결과를 생성한 후 과정이 끝이 난다[16].

2.3 ORM

ORM(Object Relational Mapping)의 등장은 Java에서 웹 애플리케이션을 개발할 때 뷰, 비즈니스 로직 및 데이터베이스 처리까지 모두 포함하여 개발하기 때문에 유지보수의 어려움이 나타나게 되었다. 따라서 MVC 모델에 입각한 각 티어를 구분하여 개발자로 하여금 비즈니스 로직에 집중할 수 있는 컴포넌트와 같은 형태들로 제공되었다. SUN에서는 J2EE 사양서에 EJB(Enterprise JavaBeans)를 채택하여 퍼시스턴스 티어(Persistence Tier)를 EJB 컨테이너에게 담당하게 함으로써 이와 같은 문제를 해결하고자 했다. 그러나 Model 2 방식에서는 각 WAS 제품에 따른 배치기술자(Deployment Descriptor) 파일을 수정해야 하고, 개발 생산성 저하와 테스트의 어려움 등의 단점이 발생하게 되었다. 또한 제조사들마다 WAS에 따른 컨테이너의 처리 방식이 다르기 때문에 EJB 뿐 만 아니라 WAS에 대한 이해도가 필요하게 되었다. 따라서 이와 같은 어려운 문제를 해결하기 위하여DAO(Data Access Object) 클래스를 두고 데이터베이스와의 처리를 DAO에서 처리하게 하였다. DAO 클래스에서는 특성에 따라 트랜잭션과 커넥션 풀 처리 등을 상이하게 개발하면 된다. 현재 ORM에서는 퍼시스턴스 티어를 처리하는 하이버네이트(Hibernate), 마이바티스(MyBatis), 톱링크(TopLink)와 같은 프레임워크가 등장하게 되었다[17].

2.4 하이버네이트와 JPA

하이버네이트는 Java 프로그래밍 언어를 위한 ORM 도구이다. 객체지향 도메인 모델(Object-Oriented Domain Model)을 관계형 데이터베이스에 매핑하기 위한 프레임워크를 제공한다. 하이버네이트는 직접적인 영구(persistent) 데이터베이스 액세스를 높은 수준의 객체 처리 기능으로 대체함으로써 객체 관계형 임퍼던스 불일치 문제를 처리한다. 하이버네이트의 주요 기능은 Java 클래스에서 데이터베이스 테이블로 매핑하고, Java 데이터 유형

에서 SQL 데이터 유형으로 매핑하는 것이다. 하이버네이트는 데이터 쿼리 및 검색 기능도 제공한다. SQL 호출을 생성하고 result set의 수동 처리 및 오브젝트 변환에서 개발자를 해방시킨다[18].

JPA는 JavaSE와 JavaEE를 사용하는 응용 프로그램에서 관계형 데이터베이스의 관리하는 API이며, 기존에 EJB에서 제공되던 엔터티 빈을 대체하는 기술이다. 또한 JPA는 Java 애플리케이션에서 관계형 데이터베이스를 사용하는 방식을 정의한 인터페이스이다. JPA는 특정 기능을 하는 라이브러리가 아니며, Java 애플리케이션에서 관계형 데이터베이스를 어떻게 사용해야 하는지를 정의하는 한 방법이다. 또한 JPA는 단순히 명세이기 때문에 구현이 없으며, 정의한 javax.persistence 패키지의 대부분은 인터페이스, enum, exception, 그리고 각종 애노테이션으로 이루어져 있다[19].

2.5 스프링 데이터 JPA

스프링 데이터(Spring Data)의 미션은 기본 데이터 저장소의 특수한 특성을 유지하면서 데이터 액세스를 위한 친숙하고 일관된 스프링 기반 프로그래밍 모델을 제공하는 것이다. 스프링 데이터는 데이터 액세스 기술, 관계형 및 비 관계형 데이터베이스, 맵 축소 프레임워크 및 클라우드 기반 데이터 서비스를 쉽게 사용할 수 있다. 이것은 주어진 데이터베이스와 관련된 많은 하위 프로젝트를 포함하는 포괄적인 프로젝트이다. 스프링 데이터 프로젝트에는 스프링 데이터 JPA, 스프링 데이터 JDBC, 스프링 데이터 REST, 스프링 데이터 MongoDB 등 많은 프로젝트들이 진행되고 있다[20].

스프링 데이터 JPA는 스프링 데이터 프로젝트에서 제공하는 모듈 중 하나로 개발자가 JPA를 더 쉽고 편하게 사용할 수 있도록 도와준다. 이는 JPA를 한 단계 추상화시킨 저장소(Repository)는 인터페이스를 제공함으로써 이루어진다. 사용자가 저장소 인터페이스에 정해진 규칙대로 메서드를 입력하면, 스프링이 알아서 해당 메서드 이름에 적합한 쿼리를 전달하는 구현체를 만들어서 빈으로 등록해준다. 스프링 데이터 JPA가 JPA를 추상화했다는 것은 스프링 데이터 JPA의 저장소의 구현에서 JPA를 사용하고 있다는 것이다[21].

Fig. 1은 스프링 데이터 JPA를 사용한 데이터베이스를 액세스하는 구조를 나타낸 것이다[22].

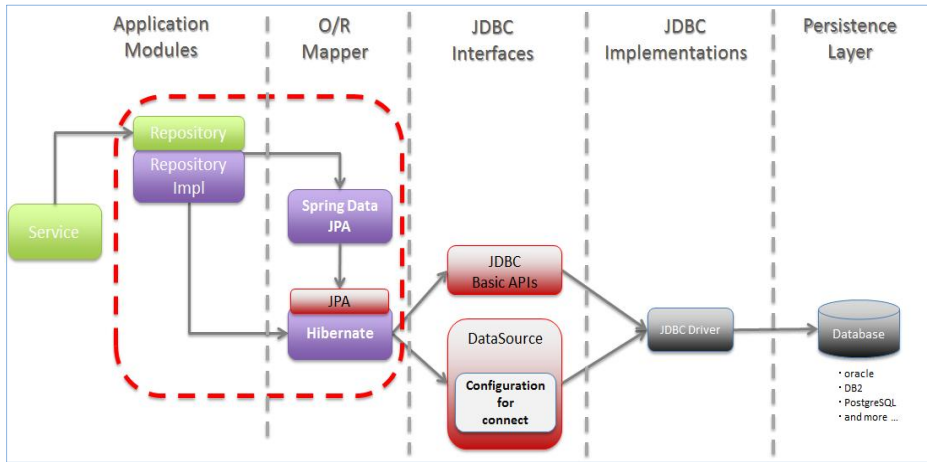


Fig. 1. DB Access Architecture with Spring Data JPA

3. 엔터프라이즈 시스템 플랫폼의 분석 및 설계

3.1 개발 환경

본 연구에서는 스프링 프레임워크 5 환경에서 스프링 데이터 JPA 기반으로 오라클 데이터베이스로 구현하기 위하여 Table 1과 같이 반려동물 관리 PetClinic을 분석하고 설계한 후 화면의 크기에 따라 웹과 모바일의 반응형 웹 디자인(RWD)으로 구현하도록 한다[23].

Table 1. Development Environment of Pilot System

Items	Contents
O/S	Windows 10 Ultimate 64bits
Web Browser	Google Chrome 78.x / IE 11
Web Container	Apache Tomcat 9.0.27
Framework	Spring Framework 5
ORM	Spring Data JPA
DBMS	Oracle 11gR2
DB Modeling	ERWin 7.3
IDE Tool	STS 4.4.2
CASE Tool	IBM Rational Rose EE 7.0

3.2 데이터베이스 스키마 설계

ORM은 객체와 테이블을 매핑해서 패러다임의 불일치 문제를 해결해 준다. 객체 측면에서 정교한 모델링을 할 수 있고, 관계형 데이터베이스는 데이터베이스에 맞게 모델링을 하면 된다. 반려동물 관리 PetClinic은 스프링 프레임워크 환경에서 다양한 엔터프라이즈 시스템 플랫폼으로 설계에 필요한 테이블로 영속적인 정보를 저장하기

위하여 데이터 모델이 먼저 정의되어야 한다. 따라서 오라클 데이터베이스 엔터프라이즈 에디션을 기반으로 데이터 모델을 작성한다. 본 연구에서 제시한 데이터 모델인 물리적 데이터베이스 스키마 구조는 Fig. 2와 같다.

3.3 요구사항 정의 활동

요구사항 정의 활동은 4+1 뷰와 연관을 시키면 유스케이스(Use Case) 관점에 해당한다. 시스템과 상호작용을 하는 사용자와 외부 연동 시스템을 액터로 파악하고 시스템에서 제공하는 각각의 기능을 유스케이스로 파악해서 이를 유스케이스 다이어그램에 표현한다[24]. Fig. 3은 PetClinic의 요구사항을 모델링한 결과를 유스케이스 다이어그램으로 표현한 모습을 보여준다.

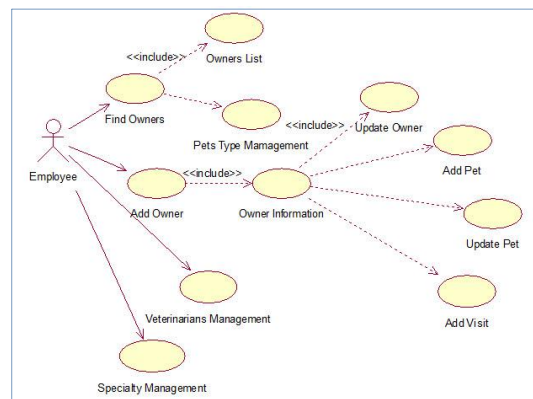


Fig. 3. Use Case Model of PetClinic

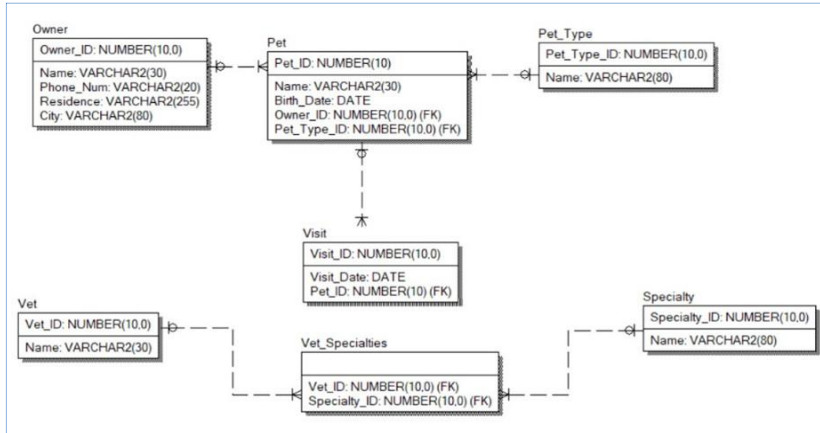


Fig. 2 Physical Model of PetClinic

유스케이스 모델과 더불어 요구사항 정의 활동의 또 다른 산출물은 유스케이스 명세서(Use Case Specification)이다. 본 연구에서는 11개의 유스케이스 별로 유스케이스 명세서를 작성하여 보다 자세한 정보를 기술하도록 하였다.

Fig. 4는 PetClinic의 화면 흐름 모델로 각 유스케이스 별로 사용자 인터페이스를 위하여 사용되는 화면들 간의 흐름 관계를 보여준다.

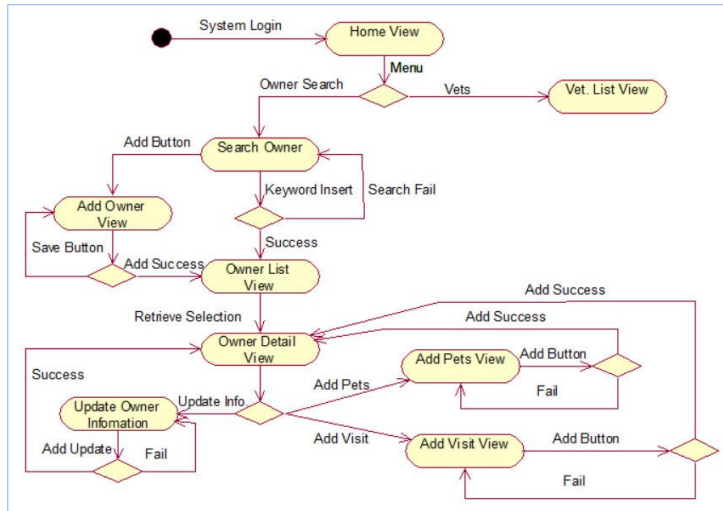


Fig. 4. View Flow Model of PetClinic

3.4 설계 유스케이스 실현 모델

설계 유스케이스 실현 모델에서는 설계활동에서 파악된 설계 클래스들이 메시지를 주고받으면서 시스템에서 필요한 요구사항을 제공할 수 있는 지를 표현한 시퀀스 다이어그램이다. 설계 유스케이스 실현 모델은 유스케이스 실현을 위하여 클래스의 객체들이 어떻게 상호작용을 하는지를 표현한다. 설계 유스케이스 실현 모델

은 시퀀스 다이어그램의 상단에 배치되는 객체가 설계 클래스의 객체이며, 시퀀스 다이어그램에 표시된 메시지는 설계 클래스에 정의된 연산에 해당된다[24]. Fig. 5는 PetClinic에서 반려동물 주인을 검색하는 설계 유스케이스 실현 모델을 시퀀스 다이어그램으로 도식한 것이다.

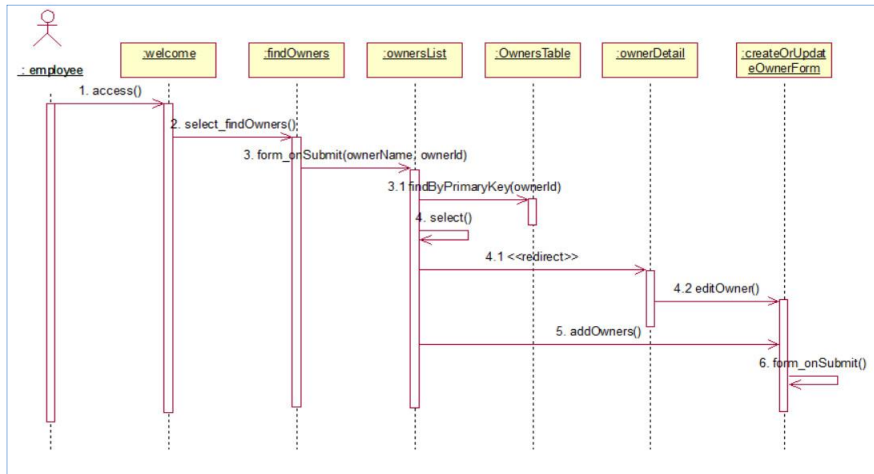


Fig. 5. Design Use Case Realization Model of Pet Owner Search

4. 스프링 데이터 JPA 기반 시스템의 구현

데이터베이스 스키마를 기반으로 스프링 프레임워크 5 환경에서 스프링 데이터 JPA 기반의 PetClinic을 구현한 웹 폼메인 화면을 보면 Fig. 6과 같다.

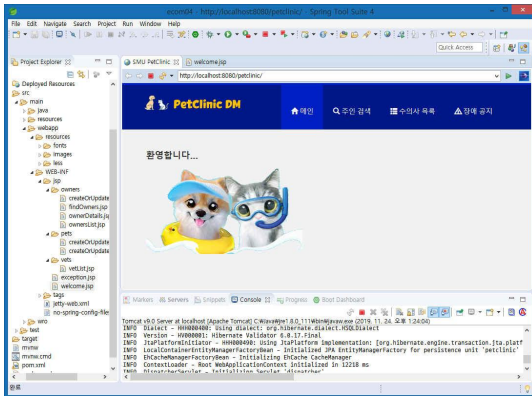


Fig. 6. Web Main Views of PetClinic

한 번의 개발로 다양한 환경을 지원하는 OSMU(One Source Muti Use) 반응형 웹 디자인으로 화면의 크기가 모바일로 되면 Fig. 7과 같이 자동적으로 변경되게 된다[25].

Fig. 8은 반려동물 주인을 검색하기 위한 화면이며, 반려동물 주인을 클릭하면 반려동물 주인의 상세 정보와 새로운 반려동물 추가가 가능하다. 또한 진료한 정보와 상세 진료 내용등도 수정과 갱신이 가능하다.

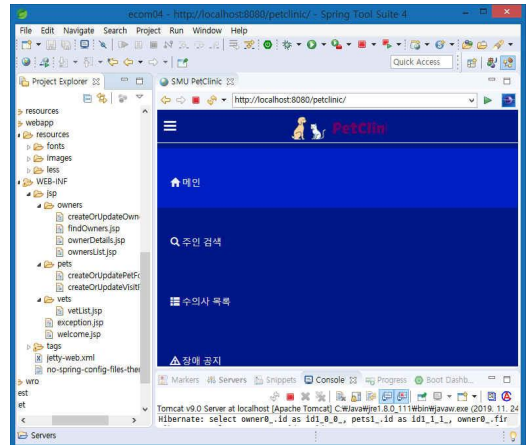


Fig. 7. Mobile Main Views of PetClinic

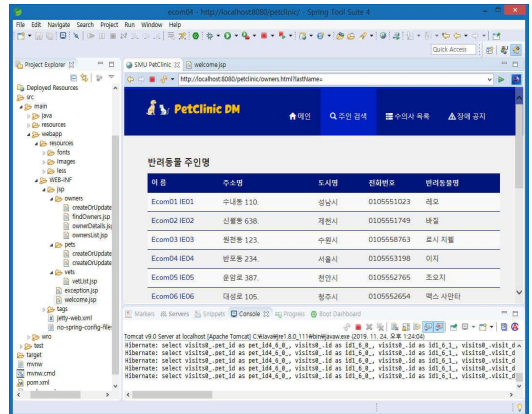


Fig. 8. Pet Owners List View of PetClinic

반려동물 수의사들의 정보와 진료과목 등을 볼 수 있는 화면은 Fig. 9와 같으며 XML이나 JSON(JavaScript Object Notation)으로 내용을 볼 수도 있다. 또한 장애가 발생한 정보도 볼 수가 있다.

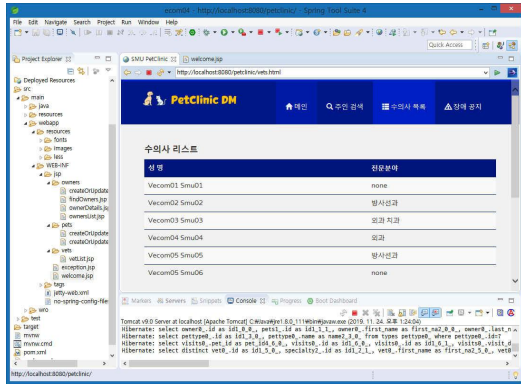


Fig. 9. Main Views of PetClinic

5. 결론

엔터프라이즈 환경의 표준화 경쟁은 백엔드의 데이터 티어로 시작하여 대표적인 엔터프라이즈 미들 티어인 WAS가 스프링 프레임워크로 받아들여짐으로써 2010년 Java EE 6 플랫폼의 표준화로 점차 안정화되고 있는 실정이다. 또한 프론트엔드에서는 고성능 클라이언트와 서버 측의 웹 애플리케이션을 개발할 수 있는 시대가 ECMAScript 6 발표에 따라 자바스크립트를 활용할 수 있게 되었다. 따라서 앞으로 출현할 수 있는 디바이스에 따라 원소스멀티유저(OSMU)가 가능한 웹과 모바일 서비스의 호환성 확보가 웹 서비스 기업들의 중요한 문제로 대두되고 있다.

그러나 몇몇 젊은 기업들을 제외하고는 현재 국내 대부분의 기업들과 대학들은 이러한 정보화 시대의 격변한 환경 변화에 따라 역량있는 인력 수급과 교육과정 개발 및 적용에 아직도 미비한 실정이다. 고무적인 현상은 올해부터 이러한 문제점을 해결하고자 우아한테크코스, 삼성 청년 SW 아카데미, 그리고 정부가 이노베이션 아카데미 재단을 설립하여 42 SEOUL로 이러한 문제들을 해결하고자 하는 노력이 일어나고 있다.

따라서 본 연구에서는 이러한 엔터프라이즈 시스템 플랫폼 환경에서 필요한 역량중심의 기술을 습득과 교육과정을 개발하기 위하여 스프링 프레임워크 5 환경에서 스프링 데이터 JPA를 활용한 시스템을 분석 및 설

계 단계별로 문서화 작성을 통하여 구현하였다.

향후 엔터프라이즈 환경에서 프론트엔드 및 백엔드의 풀 스택 기반에서 바로 적용할 수 있는 MyBatis, JPA, 스프링 데이터 JDBC, 스프링 데이터 REST 그리고 클라우드 환경 등의 다양한 개발 플랫폼으로 역량중심의 교육과정 참조 모델 연구가 지속되어야 할 것이다.

REFERENCES

- [1] Wikipedia, World Wide Web. https://en.wikipedia.org/wiki/World_Wide_Web
- [2] Wikipedia, Application Server. https://en.wikipedia.org/wiki/Application_server
- [3] Wikipedia, iPhone. <https://en.wikipedia.org/wiki/IPhone>
- [4] Luke Wroblewski. (2011). *Mobile First, A Book Apart*.
- [5] M. H. Lee. (2019). Design and Implementation of Hybrid Apps Design based on Spring MVC, *Journal of the Korea Convergence Society*, 10(3), 395-400. DOI : 10.15207/JKCS.2019.10.3.395
- [6] Wikipedia, Cloud computing. https://en.wikipedia.org/wiki/Cloud_computing
- [7] Wikipedia, Java Platform, Enterprise Edition. https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition
- [8] M. H. Lee. (2017). Design and Implementation of N-Screen Support System with Next-Generation Web Standards Framework. *Journal of the Korean Institute of Plant Engineering*, 22(2), 13-20.
- [9] M. H. Lee. (2015). A Study on N-Screen Convergence Application with Mobile WebApp Environment, *Journal of the Korea Convergence Society*, 6(2), 43-48. DOI : 10.15207/JKCS.2015.6.2.043
- [10] eGovFrame. <https://www.egovframe.go.kr/EgovDevEnvReleaseNote.jsp?menu=3&submenu=2>
- [11] Samsung SW Academy For Youth. <https://www.ssfy.com/ksf/jsp/swp/swpMain.jsp>
- [12] 42 SEOUL Innovation Academy. <https://42seoul.kr/studies?gclid=EAlaIqobChMlxCjwLqE5gIVS6mWCh0BQQ7MEAAAYASAAE>
- [13] R. Johnson. (2002). *Expert One-on-One J2EE Design and Development*, Wrox.

[14] R. Johnson, et al. (2005). *Professional Java Development with the Spring Framework*, Wrox.

[15] *Wikipedia, Spring Framework*.
https://en.wikipedia.org/wiki/Spring_Framework

[16] I. M. Lee. (2010). *Spring 3.0 of Toby*, Acorn.

[17] *Wikipedia, Object-relational mapping*.
https://en.wikipedia.org/wiki/Object-relational_mapping

[18] *Wikipedia, Hibernate(framework)*.
[https://en.wikipedia.org/wiki/Hibernate_\(framework\)](https://en.wikipedia.org/wiki/Hibernate_(framework))

[19] *Wikipedia, Java Persistence API*.
https://en.wikipedia.org/wiki/Java_Persistence_API

[20] *Spring Data*.
<https://spring.io/projects/spring-data>

[21] *Spring Data JPA*. <https://spring.io/projects/spring-data-jpa>

[22] *TERASOLUNA, Data Access (JPA)*. <https://terasolunaorg.github.io/guideline/5.1.0.RELEASE/en/ArchitectureInDetail/DataAccessJpa.html>

[23] *Wikipedia, Responsive web design*.
https://en.wikipedia.org/wiki/Responsive_web_design

[24] H. S. Chae. (2006). *Object Oriented Development Bible*, HanbitMedia.

[25] S. G. Park & H. J. Mun. (2019). Hybrid Web Content Management System Implementation. *Journal of Industrial Convergence*, 17(2), 15-20.

유 정 상(Jung Sang Yoo)

[정회원]



- 1993년 8월 : 한양대학교 산업공학과(박사)
- 2007년 9월 ~ 2008년 8월 : SIUE(Southern Illinois University Edwardsville) 방문교수

- 1996년 3월 ~ 현재 : 가천대학교 산업경영공학과 교수
- 관심분야 : 기술경영, 서비스공학, 신뢰성설계, 지능정보, TRIZ
- E-Mail : jsyou@gachon.ac.kr

이 명 호(Myeong Ho Lee)

[정회원]



- 1984년 2월 : 아주대학교 산업공학과(공학사)
- 1986년 2월 : 아주대학교 대학원 산업공학과(공학석사)
- 2001년 2월 : 아주대학교 대학원 산업공학과(공학박사)

- 2002년 3월 ~ 현재 : 세명대학교 전자상거래학과 교수
- 관심분야 : 스프링 프레임워크 Full Stack 개발, Distributed Web Programming, N-Tier Programming, RWD/AWD
- E-Mail : mhlee@semyung.ac.kr