Journal of the Korea Convergence Society
Vol. 10. No. 3, pp. 395-400, 2019

ISSN 2233-4890
https://doi.org/10.15207/JKCS.2019.10.3.395

스프링 MVC 기반에서 하이브리드 앱 디자인 설계 및 구현

이명호
세명대학교 전자상거래학과 교수

Design and Implementation of Hybrid Apps Design based on Spring MVC

Myeong-Ho Lee
Professor, Department of eCommerce, Semyung University

요 약 프론트엔드 영역의 웹 환경은 표현층의 새로운 표준을 선점하고자 하는 경쟁이 높아지고 있다. 국내에서도 다양한 기기에서 하나의 콘텐츠를 끊김없이 이용할 수 있게 해주는 서비스인 N-스크린을 미래의 핵심 서비스로 인지하여 시장 선점을 위하여 경쟁을 하고 있으며, 클라우드 컴퓨팅 시대에서 대표적인 서비스 형태가 N-스크린이다. 그러나 엔터프라이즈 환경의 그룹웨어에 필요한 프론트엔드 연구는 대부분 웹에 대해서는 반응형 웹 디자인과 모바일은 네이티브 앱에 국한되어 져 왔다. 점차 기업들의 문화적인 차이를 극복하고 하나의 소스를 여러 가지 디바이스를 지원하는 원소스 멀티유즈 전략과 개발 생산성을 위하여 MVC 디자인 패턴의 필요성은 엔터프라이즈 환경에서는 점차 확대되고 있다. 따라서 본 연구에서는 차세대 웹표준의 전자정부 표준프레임워크 환경인 스프링 MVC 기반에서 하이브리드 앱 디자인으로 JPetStore를 분석 및 설계한 후 구현함으로써 향후 엔터프라이즈 환경에서의 프론트엔드 하이브리드 앱 디자인의 참조 모델을 제공하고자 한다.

주제어 : 프론트엔드, N-스크린, 반응형 웹 디자인, 네이티브 앱, MVC 디자인 패턴, 하이브리드 앱

Abstract The Web environment of the frontend domain is increasingly competitive to preempt the new standard of presentation layer. N-Screen, a service that enables users to seamlessly use one content in various devices in Korea, is competing for market preemption by recognizing it as a core service of the future. In the cloud computing, N-screen is a typical service type. However, most of the frontend research required for groupware in enterprise environments has been limited to responsive web design for the web and native apps for mobile. Gradually, the need for MVC design patterns is increasingly widening in enterprise environments to overcome the cultural differences of companies and to support one source multi-use strategy supporting multiple devices and development productivity. Therefore, in this study, we will analyze and design JPetStore with hybrid application design based on Spring MVC, e-government standard framework environment of next generation web standard, and provide reference model of frontend hybrid apps design in future enterprise environment .

Key Words : Frontend, N-Screen, Response Web Design, Native Apps, MVC Design Patterns, Hybrid Apps

1. 서론

프론트엔드 영역의 웹 환경은 표현층의 새로운 표준을 선점하고자 하는 경쟁이 높아지고 있다. 특히 HTML5.0/CSS3 표준안이 확정됨으로써 새로운 멀티 디바이스 플랫폼도 N-tier 환경에서의 차세대 웹표준으로

*This study has been supported by the 2017 Semyung University grant.
*Corresponding Author : Myeong-Ho Lee (mhlee@semyung.ac.kr)
Received February 7, 2019 Revised February 7, 2019
Accepted March 20, 2019 Published March 28, 2019

완성되게 되었다[1]. 특히 ECMAScript의 표준화에 따라 자바스크립트를 활용한 고성능 클라이언트 측 및 서버 측의 웹 애플리케이션을 개발할 수 있는 시대가 되었다[2]. 그리고 스마트폰이 급격한 확산으로 모바일 우선주의가 웹 서비스의 핵심 전략이 되면서 모바일 환경을 지원해야 하는 필요성은 더욱 증가하게 되었다[3]. 정부에서도 차세대 웹 표준를 대비하여 스프링 프레임워크를 기반으로 하는 오픈소스를 적극 활용한 국가정보화 개발 표준 수립을 위해 전자정부 표준프레임워크를 구성하였으며, 2018년 2월 전자정부 표준프레임워크 버전 3.7.0과 12월 eGovFrame Lite 3.7.0을 발표하였다[4]. IT 관련 기업들도 다양한 디바이스에서 콘텐츠가 연속하여 서비스할 수 있게 해주는 N-스크린을 미래의 핵심 전략 서비스로 인지하고 있다[5]. 또한 웹 서비스를 제공하는 디자인 방법으로 네이티브 앱 디자인, 반응형 웹 디자인과 적응형 웹 디자인으로 크게 구분할 수 있다[6].

그러나 엔터프라이즈 환경의 그룹웨어에 필요한 프론트엔드 연구는 대부분 웹에 대해서는 반응형 웹 디자인과 모바일은 네이티브 앱 디자인에 국한되어져 왔다. 네이티브 앱 디자인은 기존 데스크톱 PC용 웹사이트와 별도로 모바일 브라우저에 최적화된 웹사이트를 추가 개발하는 전략이다. 반응형 웹 디자인은 CSS3 미디어쿼리, 유동형 그리드 레이아웃, 유연한 이미지 기법을 이용하여 페이지 레이아웃, 이미지 및 제목의 크기가 디바이스의 화면 크기에 반응하는 웹사이트 개발하는 방법이다. 적응형 웹 디자인은 반응형 웹 디자인과 클라이언트나 서버 기술을 이용하여 디바이스 해상도에 최적화된 마크업을 생성하는 방식이 결합된 기술이다[7]. 그룹웨어에서는 기업들의 문화적인 차이를 극복하고 하나의 소스를 여러 가지 디바이스를 지원하는 원소스 멀티유즈 전략에 따른 적응형 웹 디자인 연구에 대한 관심은 증가되고 있다. 또한 개발 생산성을 위하여 MVC 디자인 패턴의 필요성은 엔터프라이즈 환경에서는 점차 확대되고 있다[8].

따라서 본 연구에서는 차세대 웹표준의 전자정부 표준 프레임워크 환경인 스프링 MVC 기반에서 하이브리드 앱 디자인으로 애완동물 전자상거래 사이트인 JPetStore를 분석 및 설계한 후 구현함으로써, 향후 엔터프라이즈 환경에서의 프론트엔드 하이브리드 앱 디자인의 참조 모델을 제공하고자 한다.

2. 기존 연구에 대한 고찰

2.1 개요

웹 서비스를 개발하기 위하여 모바일 우선주의를 기반으로 할 경우, 모바일 전략과 비즈니스의 목표 에 따라 전용 모바일 OS에서만 동작하는 네이티브 앱과 모바일에 최적화되면서 디바이스 별로 별도의 개발 방식이 필요하지 않는 웹앱 그리고 앱의 장점과 웹의 기능을 혼합한 하이브리드 앱이 성공여부를 결정할 수 있다[9]. Table 1은 이러한 환경 특성에 따른 장단점을 요약한 표이다.

Table 1. Comparison Table by Development Method

Items	Advantages	Disadvantages
Native Apps	• Very fast and responsive • Best performance • More interactive, intuitive • Internet connection is not required • Overall better user experience.	• Difficult languages to learn • More expensive • Not the best option for very simple apps
Web Apps	• Easy to build • Easy to maintain • An Inexpensive option • Build one app for all platforms	• Needs a browser to run • Much slower than native apps • Web apps are less interactive and intuitive than native apps • No icon on mobile desktop • Cannot leverage device utilities
Hybrid Apps	• Built on web technology • Cheaper than a native app • One app for all platforms • No browser needed as opposed to a web app • Access to the device's internal APIs • Faster to develop than native apps	• Slower than native apps • More expensive than web apps • Less interactive than native apps • Customization will take you away from the hybrid model in which you may as well go native

2.2 반응형 웹 디자인

Ethan Marcotte에 의해 처음 소개된 반응형 웹 디자인은 하나의 웹사이트에서 다양한 디바이스에 서비스하는 디바이스 종류에 따라 화면의 크기가 자동으로 변하도록 만든 웹페이지 접근 기법을 말한다[10]. 반응형 웹 디자인은 가변적인 그리드 기반의 레이아웃, 가변적인 이미지와 미디어, 그리고 미디어 쿼리 기법을 이용한 페이지 레이아웃과 이미지와 타이틀의 크기가 디바이스의 화면 크기에 따라 반응하는 웹사이트를 개발하는 방법이

다. 모바일 웹을 탐색할 경우 Brad Frost는 반응형 디자인에서 탐색을 처리하는데 사용되는 다양한 기술을 제안하였다[11]. 따라서 디바이스의 화면 크기나 해상도에 구애받지 않고 웹 사이트의 레이아웃 구성을 나타내는 반응형 웹 구현 방식을 도식화하면 Fig. 1과 같다.

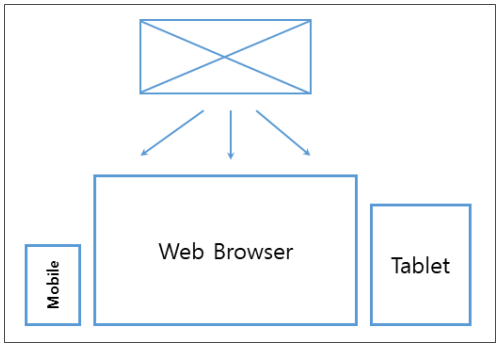

Fig. 1. View of Responsive Web

2.3 적응형 웹 디자인

Aaron Gustafson의 저서에서 처음 소개된 적응형 웹 디자인은 다른 웹 디자인 기법에서 사용하는 단일 버전과 달리 사용자 디바이스에 더 잘 맞도록 여러 버전의 웹 페이지를 만드는 것을 추구한다[7]. 적응형 웹 디자인은 그룹웨어 서비스에서 반응형 웹 디자인과 결합 할 수 있는 다양한 전략을 포함한다. 적응형 웹 디자인은 반응형 레이아웃을 포함하여 모든 유형의 웹 디자인 레이아웃을 사용할 수 있으며, 표시할 디자인 레이아웃과 크기를 선택하는 서버 사이드 검색 프로세스이다. 적응형 웹 디자인은 일반적인 디바이스 화면 크기와 해상도를 기반으로 다양한 디바이스에 사이트의 다른 버전을 제공하지만, 유일한 차이점은 레이아웃에 대한 반응형 웹 디자인의 모든 접근 방식이 아니라 부수적인 디바이스 관점에서 디자인을 검토하는 방식이다[12]. 적응형 웹의 특징은 기획 단계부터 디바이스에 맞는 해상도를 고려하기 때문에 콘텐츠 양이 많을 경우 콘텐츠의 양을 조절과 디바이스에 최적화된 디자인이 가능함에 따라 가독성을 높이는데 반응형 웹에 비해 자유롭다.

미리 정해진 몇 가지 디바이스 화면 크기를 기준으로 비율에 따라 페이지를 구현하는 적응형 웹 구현 방식을 도식화하면 Fig. 2와 같다.

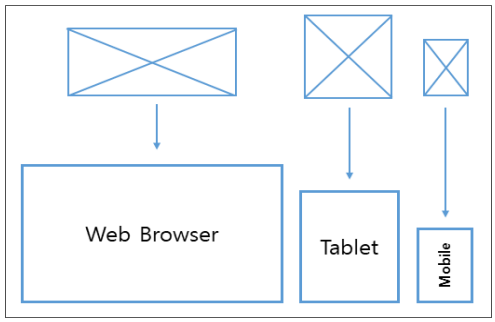

Fig. 2. View of Adaptive Web

2.4 인포그래픽 웹 디자인

방대한 양의 정보들을 최소화하면서 원하는 정보를 제공할 수 있도록 콘텐츠의 최소화로 정보의 효율성을 갖고 광고하고자 하는 것이 인포그래픽이다. 데이터의 시각화의 시초는 1786년 윌리엄 플레이페어의 저서에서 선과 막대 그리고 원 그래프를 사용해서 통계 데이터로 표현한데서 찾을 수 있다. 그 이후 오토 노이라트가 아이콘과 그림으로 개념을 설명하는 시각 소통 모델인 아이소타입을 만들면서 오늘날 인포그래픽의 아이콘 원형이라 할 수 있다[13]. 또한 인포그래픽은 정보와 자료 그리고 지식의 시각적 표현이다[9]. 현재는 동적인 대화식 데이터 기반의 시각화 라이브러리도 발표되었다[14].

2.5 스프링 MVC의 구조

MVC(Model-View-Controller) 패턴은 표현 티어와 비즈니스 티어를 분리함으로써 종속성을 줄이고 재사용성을 높여 유지보수가 가능하도록 하는 소프트웨어 디자인 패턴 방식이다[8]. 스프링 MVC는 DispatcherServlet, 핸들러 매핑, 핸들러 어댑터, 컨트롤러, 뷰 리졸버, 뷰 등 각 컴포넌트들의 역할이 명확하게 분리된다. 트랜잭션 지원과 데이터베이스 연동을 위하여 마이바티스와 스프링 컴포넌트 간의 관계 흐름을 살펴보면 Fig. 3과 같다[15].

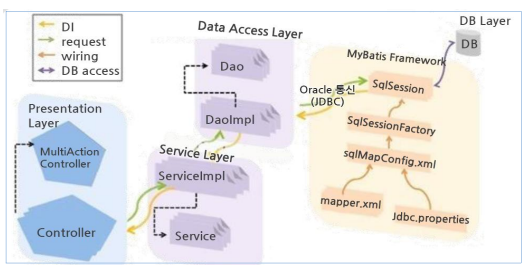

Fig. 3. Relationship of MyBatis-Spring Components

3. 하이브리드 앱의 분석 및 설계

3.1 개발 환경

본 연구에서는 스프링 MVC 기반에서 하이브리드 앱 디자인을 구현하기 위하여 Table 2와 같이 엔터프라이즈 스프링 프레임워크 환경에서 ORM 매퍼로 많이 사용되는 마이바티스를 활용하여 JPetStore를 분석하고 설계한 후 구현하도록 한다.

Table 2. Development Environment of Pilot System

Items	Contents
O/S	Windows 10 Ultimate 64bits
Web Browser	Google Chrome / MS IE 11
Web Container	Apache Tomcat 9.0.13
Framework/ Mapper	Spring Framework / MyBatis 3.x
DBMS	Oracle 11gR2
DB Modeling	ERWin 7.3
IDE Tool	STS 3.9.3(Based on Eclipse 4.7.3)
CASE Tool	IBM Rational Rose EE 7.0

3.2 데이터베이스 스키마 설계

애완동물 전자상거래 사이트인 JPetStore는 엔터프라이즈 하이브리드 앱 디자인 설계에 필요한 테이블로 영속적인 정보를 저장하기 위하여 데이터 모델이 먼저 정의되어야 한다. 따라서 오라클 데이터베이스를 기반으로 데이터 모델을 작성한다. 본 연구에서 제시한 데이터 모델인 물리적 데이터베이스 스키마 구조는 Fig. 4와 같다.

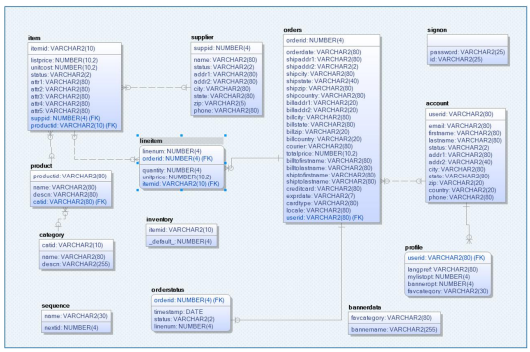

Fig. 4. Physical Model of JPetStore

4. 하이브리드 앱 시스템의 구현

데이터베이스 스키마를 기반으로 스프링 MVC 환경의 하이브리드 앱 전자상거래 JPetSotre를 구현한 메인 화면을 보면 Fig. 5와 같다.

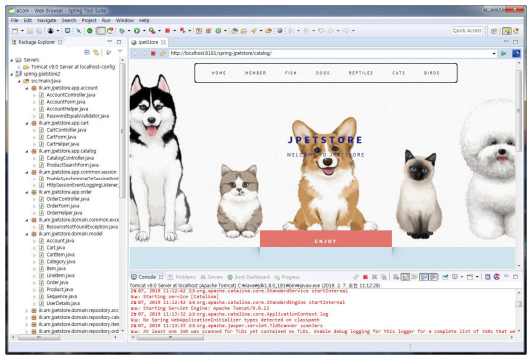

Fig. 5. Main Views of Hybrid Apps

하이브리드 앱 JPetStore 시스템에 로그인을 위한 회원가입은 해상도에 따라 웹, 테블릿 그리고 모바일에 반응적이고 적응적이게 구성한 화면은 Fig.6과 같다.

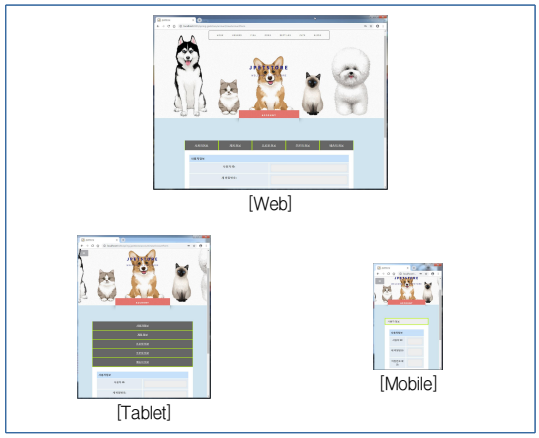

Fig. 6. Account Views of Hybrid Apps

로그인 후 애완동물별 구매를 위한 리스트 화면은 Fig.7과 같다.

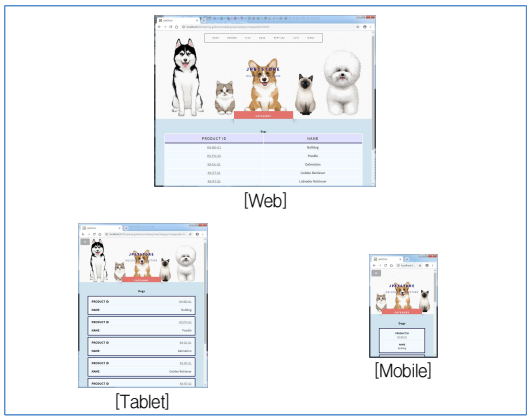

Fig. 7. Items List Views of Hybrid Apps

5. 결론

프론트엔드 영역의 웹 환경은 차세대 웹 표준의 발표와 모바일 우선 전략으로 엔터프라이즈 시스템 환경에서도 짧은 시간 동안 많은 변화의 바람이 불고 있다. 특히 ECMAScript의 표준화에 따라 자바스크립트를 활용한 고성능 클라이언트 측 및 서버 측의 웹 애플리케이션을 개발할 수 있는 시대가 도래함에 따라 오버 패칭과 언더 패칭 문제에 관심을 기울이게 되었다. 또한 다양한 디바이스의 출현에 따라 웹 및 모바일 서비스의 개발 생산성과 호환성 확보가 웹 서비스 기업들의 중요한 문제로 대두되고 있다.

그러나 엔터프라이즈 환경의 그룹웨어에 필요한 프론트엔드 연구는 대부분 웹에 대해서는 반응형 웹 디자인과 모바일은 네이티브 앱 디자인에 국한되어져 왔으며 네이티브 앱은 기존 데스크톱 PC용 웹사이트와 별도로 모바일에 최적화된 웹사이트를 추가 개발하는 전략이다. 점차 기업들의 문화적인 차이를 극복하고 하나의 소스를 여러 가지 디바이스를 지원하는 원소스 멀티유즈 전략에 따른 반응형 웹 디자인과 적응형 웹 디자인의 장점을 가진 하이브리드 앱 연구에 대한 관심이 증가되고 있다. 또한 개발 생산성을 위하여 MVC 디자인 패턴의 필요성은 엔터프라이즈 환경에서는 점차 확대되고 있다.

본 연구에서는 다양한 디바이스에 적용 가능한 엔터프라이즈 환경에서 스프링 MVC 기반 하이브리드 앱 디자인으로 전자상거래 사이트인 JPetStore를 분석 및 설계한 후 구현하여 엔터프라이즈 환경에서의 프론트엔드 하이브리드 앱 디자인의 참조 모델을 제공하였다.

향후 클라이언트 사이드의 프론트엔드에 리액트(React)와 서버 사이드의 백엔드에 그래프큐엘 서버(GraphQL Server)를 적용할 경우의 엔터프라이즈 개발 생산성에 대한 연구가 지속되어야 할 것이다.

REFERENCES

[1] MDN web docs, Modules and the standardization process, https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3
[2] Wikipedia, , ECMAScript, https://en.wikipedia.org/wiki/ECMAScript
[3] Luke Wroblewski, (2011). *Mobile First,* A Book Apart, 1-120.
[4] eGovFrame, Release Note, http://www.egovframe.go.kr/
[5] M.H. Lee and J.S. Han, (2012). Design and Implementation of JPetStore Order System Based Mobile WebApp Office, *The Society of Digital Policy & Management, 10(3),* 149-154. DOI : 10.14400/JDPM.2012.10.3.149
[6] Wikipedia, Responsive web design, https://en.wikipedia.org/wiki/Responsive_web_design
[7] Aaron Gustafson, (2011). *Adaptive Web Design,* Easy Readers, 1-116.
[8] Martin Fowler, GUI Architectures, https://martinfowler.com/eaaDev/uiArchs.html
[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] Ethan Marcotte, (2011). *Responsive Web Design,* A Book Apart, pp. 1-139.
[11] Brad Frost, responsive navigation patterns, http://bradfrost.com/blog/web/responsive-nav-patterns/
[12] Wikipedia, Adaptive web design, https://en.wikipedia.org/wiki/Adaptive_web_design
[13] Wikipedia, Otto Neurath, https://en.wikipedia.org/wiki/Otto_Neurath
[14] Mike Bostock, Data-Driven Documents, https://d3js.org/
[15] TERASOLUNA, Database Access(MyBatis3) https://terasolunaorg.github.io/guideline/5.0.1.RELEASE/en/ArchitectureInDetail/DataAccessMyBatis3.html

이 명 호(Lee, Myeong Ho) [종신회원]

- 1984년 2월 : 아주대학교 산업공학과(공학사)
- 1986년 2월 : 아주대학교 대학원 산업공학과(공학석사)
- 2001년 2월 : 아주대학교 대학원 산업공학과(공학박사)
- 2002년 3월 ~ 현재 : 세명대학교 전자상거래학과 교수
- 관심분야 : 인포그래픽, N-Tier 프로그래밍, 전자정부 표준프레임워크 Full Stack 개발
- E-Mail : mhlee@semyung.ac.kr