The Journal of the Convergence on Culture Technology (JCCT)
Vol. 7, No. 3, pp.469-474, August 31, 2021. pISSN 2384-0358, eISSN 2384-0366

http://dx.doi.org/10.17703/JCCT.2021.7.3.469
JCCT 2021-8-55

모바일 소프트웨어를 위한 효율적인 백-엔드 시스템 설계

Efficient Back-end System Design for the Mobile Software

오선진*†

Sun-Jin Oh*†

요 약 오늘날 수많은 소프트웨어 개발자들은 모바일 환경이라는 새로운 플랫폼상에서 운영되는 응용 프로그램의 효율적인 개발을 위한 백-엔드 설계 단계에 많은 어려움을 가지고 있다. 이는 큰 규모의 시스템 개발 경험도 부족할 뿐만 아니라 그것의 비구조적인 특징으로 인해 표준화된 문제 해결 방법이 없기 때문이다. 아울러, 최근의 모바일 소프트웨어 개발을 위한 시스템 설계에는 많은 도전의 중심에 큰 규모의 데이터가 있으며 이러한 데이터 중심의 모바일 응용에 대한 개발을 위해 효율적인 시스템 설계 방안이 요구된다. 본 논문은 모바일 소프트웨어가 갖는 본질적인 특징을 파악하고 많은 소프트웨어 엔지니어들이 백-엔드 설계 단계에서 겪는 어려움을 해결할 수 있는 체계적이고 효율적인 시스템 설계 방법을 제안하고자 한다.

주요어 : 모바일 플랫폼, 백-엔드 설계, 모바일 소프트웨어 개발, 시스템 설계

Abstract Nowadays, a lot of software engineers struggle with the efficient back-end design of mobile application programs operated on the new mobile platform. It is simply because not only their lack of experiences in developing large scale system but also the unstructured nature of the mobile software, where there are no standard solutions. Furthermore, since big data is at the center of many challenges in system design of mobile software, so an efficient system design scheme is required for the development of such data-intensive applications. In this paper, we propose a systematic and efficient system design method that can figure out the substantial nature of the mobile software and solve the difficulties of the back-end software engineers.

Key words : Mobile Platform, Back-end Design, Mobile Software Development, System Design

Ⅰ. 서 론

오늘날 우리들은 모바일 환경 속에서 자유롭게 이동하면서 네트워크 시스템에 접속하여 필요로 하는 서비스를 언제 어디서나 시간과 장소의 구애 없이 제공 받을 수 있는 유비쿼터스 컴퓨팅 세상을 적응하며 살고 있다. 컴퓨팅 환경도 PC 위주의 응용에서 쉽게 휴대하고 이동하기 쉬운 스마트폰과 같은 모바일 단말기기 중심의 응용으로 급속하게 변화하고 있으며 이들 단말을 위한 소프트웨어 역시 모바일기반의 소프트웨어로 발전되고 있는 상황이다. 이러한 시대적 흐름에 발맞춰 소프트웨어 개발도 모바일 본질을 갖춘 응용 앱 형태로

*종신회원, 세명대학교 정보통신학부 교수 (†교신저자)
접수일: 2021년 5월 28일, 수정완료일: 2021년 7월 15일
게재확정일: 2021년 7월 25일

Received: May 28, 2021 / Revised: July 15, 2021
Accepted: July 25, 2021
*Corresponding Author: sjoh@semyung.ac.kr
Dept. of Computer & Information Science, Semyung Univ, Korea

진화하고 있다. 하지만, 오늘날 수많은 소프트웨어 개발자들은 모바일 환경이라는 새롭게 등장한 플랫폼상에서 운영되는 응용 프로그램의 효율적인 개발을 위한 백-엔드 시스템 설계 단계에 많은 어려움을 가지고 있다. 이는 많은 소프트웨어 엔지니어들이 새롭게 선보이는 모바일 플랫폼 환경에 대한 사용과 적응준비가 잘 되어 있지 않은 측면도 있겠으나 가장 근본적인 문제는 대다수의 소프트웨어 개발자들이 이렇게 큰 규모의 시스템 개발에 경험도 부족할 뿐만 아니라 본질적으로 모바일 플랫폼 환경의 비구조적인 특징으로 인해 아직까지 이렇다 할 표준화된 문제 해결 방법이 도출되지 않았기 때문이다[1][2]. 아울러, 최근의 대다수의 모바일 소프트웨어 개발을 위한 시스템 설계 과정에서 많은 도전의 중심에 항상 큰 규모의 데이터가 연루되어 있으며, 이 이슈는 규모 확장성(scalability), 효율성(efficiency), 신뢰성(reliability), 일관성(consistency), 유지 보수성(maintainability) 등과 같은 해결하기 어려운 문제들을 동반한다. 따라서 이러한 데이터 중심의 모바일 응용에 대한 체계적인 소프트웨어 개발을 위해 효율적인 시스템 설계 방안이 절실하게 요구된다[3].

요즘 대중들로부터 가장 각광을 받으며 널리 사용되고 있는 대표적인 모바일 소프트웨어로는 카카오톡(Kakao Talk), 인스타그램(Instagram), 우버(Uber), 페이스북(Facebook Messenger), 트위터(Twitter), 넷플릭스(Netflix), 유투브(Youtube), 드롭박스(Dropbox) 등을 꼽을 수 있으며 이들 모바일 소프트웨어는 수많은 사용자를 보유하고 있고 이들로부터 시시각각으로 발생시켜 처리해야 하는 데이터의 양도 어마어마하다고 할 수 있다[4]. 이러한 모바일 사용 환경을 가지고 있는 모바일 소프트웨어의 시스템 디자인을 효율적으로 하기 위해선 컴퓨팅 장비, 데이터베이스, 네트워크 환경, 저장장치, 시스템 성능 및 보안 등 고려해야 할 사항들이 너무나도 많다[5]. 따라서 이러한 큰 규모의 모바일 소프트웨어를 성공적으로 구현하기 위해서는 우선 소프트웨어 개발 엔지니어들의 체계적이고 효과적인 시스템 설계 과정이 선행되어야 한다. 본 논문에서는 이러한 모바일 소프트웨어가 갖는 본질적인 고유의 특징을 정확히 인식하고 성공적인 모바일 소프트웨어 구현을 위해 많은 소프트웨어 엔지니어들이 백-엔드 시스템 설계 단계에서 겪는 문제점을 해결할 수 있는 체계적이고 효율적인 시스템 설계 방법을 제안하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 본 연구를 하게 된 연구 배경과 연구의 필요성을 서술하였고, 3장에서는 본 논문에서 제안하는 모바일 소프트웨어 구현을 위한 체계적이고 효율적인 백-엔드 시스템 설계 방법을 제안하였으며, 4장에서는 제안한 시스템 설계 방법을 이용한 실무사례와 결과를 고찰하였고, 마지막으로 5장에서 향후 연구과제와 함께 결론을 맺는다.

II. 연구 배경

시스템 설계 과정이란 소프트웨어 개발자가 응용 소프트웨어를 계획하여 구현하고자 할 때 이를 운영하는 컴퓨터 시스템의 정의와 요구 분석 결과를 바탕으로 시스템이 이러한 요구사항을 충족시킬 수 있도록 하드웨어나 소프트웨어의 전체적 구조, 구성요소, 모듈, 인터페이스 등을 규정하는 작업 과정으로 어떤 응용 프로그램을 컴퓨터를 이용하여 수행하려면 시스템 분석, 시스템 설계, 프로그램 설계, 프로그램 작성과 테스트, 실제 운용 등 여러 가지 작업이 필요하게 되는데 이를 시스템 설계 과정이라 정의한다[6]. 여기서 하드웨어 및 소프트웨어의 구성, 운용 방법과 체제, 이용자가 불편 없이 사용할 수 있도록 하는 것 등이 시스템 설계의 주요 내용이며, 소프트웨어의 명세를 결정하고 기술할 부분을 지시하고, 코드 설계, 입출력 설계, 파일 설계 등의 작업도 포함한다[7]. 이러한 시스템 설계는 소프트웨어 엔지니어들 중에서 프로그램 개발업무에 최소한 5년 이상의 경력을 가지고 있는 자가 담당하며 이들을 통상 Software Architect라 한다.

시스템 설계는 그 수행하는 업무의 특징에 따라 front-end와 back-end 시스템 설계로 분류할 수 있는데 소프트웨어 아키텍처에서 하드웨어와 최종 사용자 사이에는 많은 레이어로 구성되어 있다. 이때 front-end는 사용자 친화적인 인터페이스를 제공함으로써 기저에 관련된 구성요소들을 단순화시키는 추상화 개념이며 사용자에게 쉽고 편리하게 제공되는 서비스로 간주되는 반면, back-end는 일반적으로 서비스 제공을 지원하는 인프라 구조인 데이터 저장소나 비즈니스 로직을 말한다. 따라서 front-end 시스템 설계는 주로 사용자가 시스템에 편리하게 접근하여 사용할 수 있는 UI 디자인에 집중되는 반면, back-end 시스템 설계는 이러한 서비스를 제공하는 서버 디자인에 초점이 맞춰져 있다

[8]. 오늘날 주류를 이루는 모바일 소프트웨어에 대한 시스템 설계는 다수의 사용자를 대상으로 큰 규모의 데이터를 신속하고 정확하게 처리해야 하는 컴퓨팅 장비와 저장장치, 그리고 시스템 사용 환경에서 사용자의 이동성을 보장해야 하는 무선 기반의 통신과 모바일 컴퓨팅 네트워크 및 클라우드 환경 등 모바일 특성을 가진 서버 시스템에 대한 back-end 설계가 사용자들에게 주로 편이성을 제공하는데 중점을 두는 front-end 시스템 설계만큼이나 중요하며, 이를 체계적이고 효율적으로 개발하고 처리할 수 있는 back-end 시스템 설계가 절실히 요구된다.

III. 백-엔드 시스템 설계 방법

이 장에서는 본 논문에서 제안하는 체계적이고 효율적인 모바일 소프트웨어 개발을 위한 백-엔드 시스템 디자인 방법을 제안한다. 다음의 그림 1은 본 논문에서 제안하는 체계적이고 효율적인 백-엔드 시스템 설계 과정을 단계별로 보여준다.

그림 1. 백-엔드 시스템 설계
Figure 1. The Back-end System Design

그림 1에서 보인 바와 같이 백-엔드 시스템 설계 과정은 크게 7개 단계로 나누어 생각할 수 있다. 첫 번째 단계는 요구사항 분석과 시스템의 목표 설정이다. 소프트웨어 개발을 통해서 우리가 풀어야 할 문제의 정확한 범위를 설정하는 것이 소프트웨어 구현에 앞서 언제나 현명한 아이디어이다. 하지만 여기에는 정확한 답이 있는 것은 아니고 대부분 여러 방안들이 있을 수 있으며, 소프트웨어를 설계하는 단계에서 시스템의 최종 목표를 정의하는데 많은 시간을 들일수록 성공할 확률이 높아진다. 두 번째 단계는 시스템 디자인 고려사항 파악 단계로 현재 개발하고 있는 소프트웨어가 다른 소프트웨어와는 달리 특별히 요구되는 고려사항들이 있는지를 본격적으로 개발하기 전 단계에서 미리 점검하고 식별해 두는 것이 바람직하다. 그렇지 않으면 개발이 진행되고 나서 뒤에 특별한 요구사항이 발견되어 지금까지 개발했던 작업이 수포로 돌아가고 처음부터 다시 시작해야 하는 경우가 발생할 수 있기 때문이다. 다음 단계는 개발한 시스템이 감당해야 하는 용량의 예측평가와 제약사항(capacity estimation and constraints) 도출 단계로 개발한 소프트웨어가 운용될 때 시스템이 감당해야 하는 트래픽, 저장장치, 대역폭, 메모리 등의 예측평가와 이들 자원들의 사용에 있어 제약이 되는 사항들이 있는지 인식하고 도출해 내는 작업을 수행하게 된다. 우리가 설계하는 시스템의 규모를 예측 평가해 두는 것이 항상 좋은 생각이다. 그 이유는 이러한 작업들이 시스템의 스케일링(scaling), 분할(partitioning), 캐싱(caching) 그리고 부하 균등화(load balancing) 등을 고려하는 단계에서 많은 도움이 되기 때문이다. 그 다음 단계는 개발하는 소프트웨어에서 사용할 시스템 인터페이스를 정의하고 이때 사용할 시스템으로부터 기대되는 주요 함수 API와 parameter, 그리고 이들로부터 예상되는 반환값 등을 정의하는 단계이다. 이 단계에서는 시스템으로부터 예견되는 정확한 데이터 교환 접점을 설립해 줄 뿐 아니라 우리가 어떤 시스템 요구사항들을 틀리게 도출해서 사용하고 있는지를 확인시켜주게 된다. 다섯 번째 단계는 시스템에서 사용할 데이터 모델을 정의하는 단계이다. 일찍이 시스템에서 사용할 데이터 모델을 정의함으로써 시스템의 서로 다른 구성요소들 사이에서 데이터가 어떻게 흐르는가를 명확하게 해주며, 후에 데이터 분할과 관리하는데 가이드 역할을 하게 된다. 아울러, 어떤 형태의 데이터베이스 시스템을 사용해야 할지, DB 질의어는 어떤 것을 쓸지, 데이터를 저장할 블록 저장장치는 어떤 종류로 할지 등을 결정하게 된다. 여섯 번째 단계는 본격적인 시스템 설계를 하는 단계로 High level 설계와 Detailed 설계 단계로 나눌 수 있다. High level 디자인에서는 개발하

고 있는 시스템의 코어 컴포넌트를 나타내는 7개 내외의 box로 구성된 블록 다이어그램을 그려 디자인하게 되는데 이때 우리는 시스템의 실제 문제들을 해결하는데 필요한 충분한 컴포넌트를 식별해 내야 한다. 상세 설계 단계에서는 식별된 컴포넌트들에 대한 2-3 레벨 깊은 상세한 기능들을 식별하고 설계하는 과정이다. 마지막으로 일곱 번째 단계는 시스템 유지보수 차원의 문제 도출과 해결방안 모색 단계로 소프트웨어의 사용과 시스템 운영 중 발생할 수 있는 예기치 않은 문제들을 찾아내고 해결하는 과정이라 할 수 있다. 예를 들어, 시스템의 병목현상 발견과 해결방안, 중요한 데이터의 분리와 복사 방안, 부하 균등화 방안, 데이터 제거와 데이터베이스 청소작업, 시스템의 신뢰도와 데이터의 중복 방안, 응용의 신뢰도와 처리 속도 향상을 위한 데이터 쪼개기 방안, 보안과 승인 방안 등을 포함한다.

IV. 사례연구 및 고찰

이 장에서는 본 논문에서 제안한 체계적이고 효율적인 백-엔드 시스템 설계 방법을 요즘 실제 많이 사용되고 있는 모바일 소프트웨어인 인스타그램의 개발 단계에서의 백-엔드 시스템 설계 과정에 적용한 사례를 중심으로 고찰하고자 한다. 여기서 우리는 사례연구를 위해 단순 버전의 인스타그램 설계를 고려한다. 본 논문에서 고려한 인스타그램의 주요 기능은 사용자가 그들의 사진이나 비디오를 업로드하여 다른 사용자와 공유할 수 있도록 하는 소셜 네트워킹 서비스를 제공한다.

1. 요구사항과 시스템의 목표

1) 기능적/비기능적 요구사항 분석

사용자는 사진이나 비디오를 업로드/다운로드/보기 기능을 할 수 있으며 그 제목을 가지고 검색할 수 있다.

서비스는 항상 가용하고 시스템의 허용 지연시간은 300ms 이하이고 업로드 된 사진이나 비디오는 절대로 분실될 수 없다.

2. 시스템 디자인 고려사항

1) 시스템의 특이사항

인스타그램은 사진들을 빠른 속도로 검색할 수 있는데 focus가 맞춰진 주로 읽기 기능 중심으로 수행.

실제 사용자들은 그들이 선호하는 많은 사진을 업로드 할 수 있어 효율적인 저장장치 관리가 시스템 디자인에 있어 가장 중요한 요소.

3. 시스템 용량 예측평가와 제약사항

1) 소프트웨어 사용 환경 분석을 통해 미리 트래픽, 저장장치, 대역폭, 메모리 등의 예측평가 수행.

다음의 그림 2는 예를 든 인스타그램 사례에 대한 사용 환경 분석을 통한 네트워크 트래픽, 저장장치, 데이터 송수신 대역폭, 캐시 메모리 등 시스템 용량 예측평가를 한 결과 일부를 보여준다.

<인스타그램 사용 환경 분석>

총 가입자 수 : TM명
1일 평균 접속자 수 : DM명
1일 평균 새 사진 등록 수 : Dphoto개
사진 평균 사이즈 : 200KB

[Traffic Estimates]
QPS(초당 질의 수)
<쓰기작업> DM명 / (24시간 X 60분 X 60초) 회
<읽기작업> 인스타그램은 읽기 위주 중심으로 (R/W비 = 100:1)
100 X DM명 / (24시간 X 60분 X 60초) 회

[Storage Estimates]
인스타그램을 5년간 운영하고 시스템을 update할 예정이라면
사진의 1일 요구 storage space : Dphoto X 200KB
5년간 요구되는 총 storage space : Dphoto X 200KB X 365일 X 5년

[Bandwidth Estimates]
<쓰기작업> 초당 서비스로 들어오는 총 데이터 크기
= DM명 / (24시간 X 60분 X 60초) 회 X 200KB (KB/sec)
<읽기작업> 초당 서비스로 나가는 총 데이터의 크기
= 100 X DM명 / (24시간 X 60분 X 60초) 회 X 200KB (KB/sec)

[Memory Estimates]
빈번하게 접근하는 작업에 대해 Cache 메모리 사용
80-20 rule 적용 (즉, 20% 작업이 80%의 트래픽 유발한다는 규칙)
여기서 이 20% hot한 작업에 대해 cache 사용한다면
하루 당 예상 총 접속 질의 수는
= QPS(read) X 60초 X 60분 X 24시간
이 중 hot한 접속 질의만이 Cache를 사용하므로
= QPS(read) X 60초 X 60분 X 24시간 X 20% (KB)
단, 여기서 많은 접속 질의가 중복될 수 있으므로 실제 Cache 사용량은 이 보다는 작다.

<중략>

그림 2. 시스템 용량 예측 평가
Figure 2. System Capacity Estimation

4. 시스템 인터페이스와 주요 API 함수 정의

이 단계에서는 개발되는 소프트웨어에서 사용할 시스템 인터페이스를 정의하고, 이때 시스템으로부터 기대되는 주요함수 API와 parameter, 예상 반환값 등을 정의한다. 이 단계에서는 시스템으로부터 예견되는 정확한 데이터 교환 접점을 설립해 주며 또한 우리가 틀리게 도출해서 사용하고 있는 시스템 요구사항들을 확인시켜주게 된다.

5. 데이터 모델 정의

시스템에서 사용할 데이터의 유형, 구성방법, 사이즈

등을 정의하는 단계로 예를 들어 인스타그램에서의 사용자 데이터의 경우 userID(4byte) + name(20byte) + address(20byte) + … 등으로 정의하며, photo의 경우에는 photoID(4byte) + userID(4byte) + … 등으로 정의하게 된다.

6. High level 시스템 설계와 상세 디자인

이 단계에서는 시스템 설계를 top-down 방식으로 high level 시스템 설계에서, database schema 정의, 주요 컴포넌트 설계, detailed design 순으로 진행한다. 다음의 그림 3은 예를 든 인스타그램 사례에 대한 High level 시스템 설계를 한 결과를 보여준다.

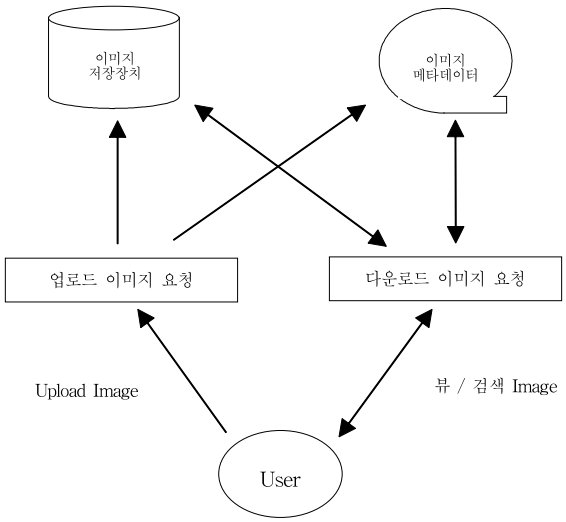

그림 3. 하이레벨 시스템 설계
Figure 3. High Level System Design

7. 시스템 유지보수 차원의 문제 도출과 해결방안

이 단계는 시스템 유지보수 차원의 문제 도출과 해결방안 모색 단계로 소프트웨어의 사용과 시스템 운영 중 발생할 수 있는 예기치 않은 문제들을 찾아내고 해결하는 과정이라 할 수 있다.

<시스템 운영 중 성능이나 효율성에 영향을 미치는 고려사항>
- Identifying & resolving bottlenecks
- Data partitioning & replication (Data sharding)
- Cache capacity estimation & Load balancing
- Purging or DB cleanup
- Telemetry
- Security & permissions
- Reliability & redundancy

<중략>

그림 4. 시스템 성능과 효율성을 위한 고려사항
Figure 4. Consideration for System Performance & Efficiency

그림 4는 시스템을 운영하면서 발생할 수 있는 시스템 성능이나 효율에 영향을 미칠 수 있는 문제 해결을 위한 주요 고려사항 관련 내용을 포함한다. 이 작업은 추후 시스템 운영 중 발생할 수 있는 유지보수 비용에 직접적인 영향을 미친다.

V. 결 론

최근 많은 소프트웨어 개발자들은 새로운 모바일 환경에서 동작하는 모바일 소프트웨어 구현을 위한 백-엔드 시스템 설계에 많은 어려움을 토로한다. 그 이유는 큰 규모 시스템 개발 경험도 부족할 뿐 아니라 그것의 비구조적인 성향으로 인해 표준화된 문제 해결 방법이 없기 때문이다. 이러한 큰 규모의 데이터와 연관된 데이터 중심 모바일 소프트웨어의 개발에는 단계적이고 효율적인 백-엔드 시스템 설계 방안이 요구된다. 본 논문에서는 이러한 모바일 소프트웨어가 갖는 본질적인 특징을 파악하고 많은 소프트웨어 엔지니어들이 백-엔드 설계 단계에서 겪는 어려움을 해결할 수 있는 체계적이고 효율적인 백-엔드 시스템 설계 방법을 제안하고 사례연구를 통해 고찰하였다. 향후 연구과제는 제안한 백-엔드 시스템 설계 방법을 이용한 모바일 응용 구현에 관한 것이다.

References

[1] M. B. Dishman and M. Fowler: “Agile Architecture,” O’Reilly Software Architecture Conference, March 2015.
[2] S. Oh, “Design of the Smart Application based on IoT”, Journal of IIBC, Vol. 17, No. 5, pp. 151-155, October 2017. DOI: https://doi.org/10.7236/JIIBC.2017.17.5.151
[3] M. Kleppmann, Designing Data-Intensive Applications : The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, O'Reilly, pp. 590, 2017.
[4] B. Schneier, Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World. Norton, 2015.
[5] N. Marz, J. Warren, Big Data : Principles and Best Practices of Scalable Real-Time Data Systems. Manning, 2015.
[6] http://joomlatuts.net/joomla-2-5/87-how-backend-model-view-controller-mvc-works-in-joomla/98-difference-between-frontend-and-backend-mvc

[7] https://techdocs.altium.com//display/ADOH/Front-End+Design
[8] http://research.cs.tamu.edu/prism/lectures/sp/l18.pdf