

# 통신 설비 구축 및 유지보수 업무의 프로세스 개선과 데이터 축적을 위한 모바일 웹 설계

김효준\*, 유철중\*\*

## Mobile Web Design for Process Improvement and Data Accumulation in Communication Infrastructure Establishment and Maintenance

Hyo-Jun Kim\*, Cheol-Jung Yoo\*\*

### 요 약

기업의 경쟁력을 강화하는 데 고려해야 할 사항 중 하나로 업무 프로세스를 빼놓을 수 없다. 비효율적인 업무 프로세스는 업무 생산성 저하로 기업의 경쟁력에 악영향을 끼치기 때문이다. 또한 최근 빅데이터와 빅데이터를 기반으로 한 AI 등의 데이터 중심 산업 등 데이터 관련 산업이 주목받으면서 기업의 자체 데이터 축적 역시 기업의 경쟁력 강화의 주요 요소로 떠오르고 있다. 본 논문에서는 비효율적인 업무 프로세스를 개선하고, 업무 프로세스의 효율을 저하하지 않는 선에서 미비한 자체 데이터 축적 문제를 해결하기 위한 모바일 웹 시스템을 UML(Unified Modeling Language)을 사용하여 객체지향 설계한다. 또한 시스템을 스프링 프레임워크(Spring framework)로 구현하는 것을 목표로 설계하여 구현 단계에서의 이해를 도우며, 데이터 축적을 위해 실제 업무에서 사용되는 용어와 데이터를 기반으로 데이터베이스 E-R 다이어그램(Entity relationship diagram)을 작성한다.

### Abstract

One of the factors that must be considered to enhance a company's competitiveness is the business process. Inefficient business processes can negatively impact productivity and undermine a company's competitiveness. Additionally, with the recent prominence of data-driven industries such as big data and AI, the accumulation of internal data has emerged as a key element in enhancing a company's competitiveness. In this paper, we aim to design a mobile web system using Unified Modeling Language to improve inefficient business processes and address the issue of insufficient internal data accumulation without compromising the efficiency of the business process. Furthermore, the goal is to implement the system using the Spring framework to facilitate understanding during the implementation phase. To facilitate data accumulation, an Entity-Relationship diagram will be created based on the terminology and data used in actual business operations.

### Keywords

business process, data accumulation, object oriented design, UML, Spring framework

\* 전북대학교 융합기술경영학과 석사과정

- <https://orcid.org/0009-0007-1539-4531>

\*\* 전북대학교 소프트웨어공학과 교수(교신저자)

- <https://orcid.org/0000-0002-7903-2309>

• Received: Jul. 18, 2023, Revised: Sep. 11, 2023, Accepted: Sep. 14, 2023

• Corresponding Author: Cheol-Jung Yoo

Dept. of Software Engineering, Jeonbuk National University, Korea

Tel.: +82-63-270-3383, Email: [cjyoo@jbnu.ac.kr](mailto:cjyoo@jbnu.ac.kr)

## 1. 서 론

### 1.1 연구의 배경

기업의 경쟁력을 강화하는 데 고려해야 할 사항 중 하나로 업무 프로세스를 빼놓을 수 없다. 업무 프로세스가 비효율적이라면 한 건의 업무를 수행하는 데 필요한 시간이 길어져 업무 담당자들의 업무 생산성이 저하되고, 이는 기업 경쟁력에 악영향을 끼치기 때문이다. 또한 최근 빅데이터와 빅데이터 기반 산업이 주목받음에 따라 기업의 자체 데이터 축적 역시 기업의 경쟁력 강화의 주요 요소로 떠오르고 있다. 빅데이터는 우리의 삶, 업무, 사고방식을 변화시킬 혁명적인 움직임으로, 이전에 경험하지 못한 규모와 다양성으로 생성되는 디지털 정보의 방대한 모음을 의미한다[1]. 현재 빅데이터 분석 분야에서는 한 영역의 데이터뿐만 아니라, 다른 영역의 데이터를 서로 결합하고 분석하여 새로운 가치를 창출해내기도 한다. 기업에서는 빅데이터를 활용하여 기업의 가치사슬에 미치는 영향을 분석하고, 기업 경쟁력 강화의 발판으로 삼기도 한다. 가치사슬은 1985년 마이클 포터 교수가 처음 제안한 개념으로, 기업에서 제품을 생산하거나 서비스를 제공하기 위해 자원을 통합하는 일련의 절차를 지칭한다[2]. 빅데이터 활용이 기업의 가치사슬에 미치는 영향으로는 빅데이터 분석을 활용하여 생산 계획을 구체화하거나 적절한 시간과 장소에 원자재를 공급하거나 비효율적인 업무 프로세스를 재설계하는 등의 생산 활동에서 가장 큰 효과를 얻고 있는 것으로 확인되었다[3]. 데이터는 기업에 있어서 새로운 가치를 창출해낼 수도 있고 기업의 업무 프로세스에도 기여할 수 있는 유의미한 자산이며, 이로 인해 기업의 자체 데이터 축적은 날이 갈수록 중요해지고 있다. 그러나 기업에서 자체적으로 데이터를 축적할 수 있는 시스템을 갖추는 것은 쉬운 일이 아니며, 이로 인해 데이터 축적을 시스템 갖추고 축적 데이터를 활용하는 기업과 기업 경쟁력의 격차가 벌어지고 있다.

### 1.2 연구의 필요성 및 목적

기업의 입장에서 데이터 축적을 시작하기 위한 방안을 마련하는 것은 쉬운 일이 아니다. 먼저, 데이터를 축적하기 시작한다고 해서 당장 유의미한 빅데이터가 되는 것이 아니다. 앞서 언급한 대로 빅데이터는 의미 그대로 많은 양의 데이터이다. 어느 정도 양의 데이터가 축적되기까지는 당연하게도 시간이 필요하다. 짧은 기간 동안 축적된 빈약한 데이터를 활용하여 어떠한 분석이나 관련 데이터의 결합 등으로 가치를 창출해 낼 수는 없기 때문에 긴 시간 동안 데이터를 수집할 필요성이 있다.

그러나 실제 업무에서 데이터 축적을 시작하여 오랜 기간 지속하는 방안을 마련하는 것은 까다로운 일이다. 데이터를 축적할 방법을 구상하여 이를 업무 프로세스에 추가하고, 해당 프로세스를 반복하면 되는 단순한 문제가 아니다. 실제 업무에서 업무 프로세스는 업무의 효율성을 결정짓는 중요한 요소다. 업무 프로세스에서 데이터를 축적하기 위한 프로세스가 추가된다면 당연하게도 프로세스가 복잡해지고, 그 결과 하나의 업무 프로세스 사이클이 완료되기까지 걸리는 시간이 길어져 업무의 효율이 떨어진다. 또한 유의미한 데이터 항목을 결정하는 것도 중요한 업무라고 할 수 있다. 축적할 데이터 항목을 선별하여 축적된 데이터를 활용하기 위한 데이터 정제 작업에 기여할 수 있기 때문이다.

이에 따라 본 논문에서는 기존 업무 프로세스를 분석한 후, 업무 프로세스의 개선과 함께 개선된 프로세스상에서 데이터를 축적하는 방안을 제시하고, 개선 방안을 바탕으로 한 업무 시스템을 설계하고자 한다.

## II. 관련 연구

비즈니스 프로세스와 기업의 자체 데이터 축적 모두 그 중요성이 강조되고 있는 분야인 만큼 관련 연구 또한 활발히 진행되고 있다.

먼저 비즈니스 프로세스 분야를 살펴보면, [4]에서는 건설업계에서의 설계 관리 업무를 분석하고 실제 설계 정보 표본을 통해 표준화된 정보 중심의 설계 업무 프로세스를 제안하였다. [5]에서는 프로세스 가상화 이론에 기반하여 가상 환경 업무 프로

세스에 업무 특성(관계, 감각 등)과 IT 특성(업무 IT 환경 및 가상화 도구), 조직 특성이 어떠한 영향을 끼치는지에 대해 연구하였다. [6]에서는 업무 프로세스 개선의 중요성을 언급하고, 프로세스 단순화 등으로 대표되는 재설계 패턴에 기반한 업무 프로세스의 개선 방안인 비즈니스 프로세스 재설계(Business process redesign)의 개념화를 제안하였다. [7]에서는 비즈니스 프로세스에서 예외적인 상황에 대처하기 쉽도록 재설계 매트릭스를 제공하는 프로세스 재설계 프레임워크(Framework)를 제안하였다. [8]에서는 비즈니스 프로세스를 효율적으로 관리하는 것을 돕기 위한 프로세스 모델링 프레임워크를 제안하였다.

다음으로 기업의 자체 데이터 관련 연구를 살펴보면, [9]에서는 기업의 현재 사용 기록에 한정하여 경영적인 측면에서의 데이터 축적 필요성 및 그에 따른 이익과 빅데이터 측면에서의 효율성을 연구하였다. [10]에서는 산업 분류, 전문 인력 부족 등의 이유로 데이터 수집에 어려움을 겪는 기업들을 위해 각 산업 분야의 특정 업무 프로세스에 맞춤형 애플리케이션 개발을 연구하여 해결책을 제시하였다. [11]에서는 데이터 기반 자본주의를 언급하며, 데이터 축적, 추출의 중요성을 강조하였다. [12]에서는 알루미늄 합금 빌렛 제조 공정 데이터를 수집 및 관리하는 플랫폼을 구현하고, 축적된 데이터와 기상청 제공의 공공 데이터를 활용하여 유의미한 통계 자료를 추출하는 등 자체 데이터 활용 모델을 제시하였다.

이와 같이 많은 관련 연구에서 비즈니스 프로세스와 기업의 자체 데이터 축적의 중요성을 강조하고 있으며, 두 분야 모두 기업 입장에서 개선 및 관리해야 하는 중요한 부분이라고 할 수 있다.

비효율적인 업무 프로세스는 기업의 생산성을 저하시키는 요소이며, 기업의 자체 데이터는 업무 프로세스 재설계 등에 활용하거나 빅데이터 산업 진출을 통한 기업 경쟁력을 강화하는 중요한 요소이다.

따라서 본 논문에서는 기업의 비효율적인 비즈니스 프로세스의 개선과 더불어 프로세스의 흐름을 방해하지 않는 데이터 축적을 위한 모바일 웹 시스템을 설계하고자 한다.

### III. 통신 설비 구축 업무 프로세스 분석 및 개선 방안

#### 3.1 현황 분석

연구 표본으로 선정된 K 기업은 통신 설비 시공 업체로, 큰 틀에서 건설업에 해당하며 건물, 전신주, 통신 맨홀 등에 인터넷 및 모바일 망을 구축하는 것을 주 업무로 하는 통신사의 협력 업체이다. K 기업의 업무 프로세스에는 통신 설비 구축 및 유지보수 공사와 해당 공사를 설계하는 작업도 포함한다.

##### 3.1.1 업무 프로세스

연구 표본으로 선정된 K 기업에서 수행하고 있는 업무 프로세스의 문제점을 파악하기 위해 K 기업의 주 업무인 통신 설비 구축 및 유지보수 프로세스 중 설계자와 작업자 사이의 업무 프로세스 현황을 분석한다. 설계자는 공사를 설계하는 담당자이며, 작업자는 설계자로부터 공사 관련 서류를 받고 해당 서류를 바탕으로 실제 현장에서 공사 작업을 수행하는 근로자이다.

그림 1은 설계자의 공사 설계 완료 후, 설계자와 작업자 사이의 간략한 업무 프로세스를 보여주고 있다. 그림 1의 업무 프로세스상에서 파악되는 가장 큰 문제점은 설계자와 작업자 간의 협업 수단이다. 설계자 측에서는 설계 상세 내용을 구두 설명 및 실물 서류에 수기 작성하는 방식으로 작업자에게 전달하고 있으며, 작업자 측에서도 작업 사진과 같은 이미지 파일을 제외하면 실물 서류에 작업 상세 내역을 수기로 작성하여 설계자에게 전달하고 있다. 이렇듯 업무 담당자들 간의 비효율적인 업무 내용 전달은 커뮤니케이션 오류 등의 문제로 업무 효율에 악영향을 끼칠 수 있다.

##### 3.1.2 데이터 축적

이 절에서는 업무 프로세스에서의 데이터 축적 현황을 분석한다. 앞서 분석한 공사 설계에서부터 공사 완료까지의 프로세스에서는 실물 서류를 통해 데이터를 주고받아 데이터 축적이 불가능한 것으로

확인되었기에, 해당 프로세스 완료 후에 수행되는 정산 업무 프로세스를 분석한다. 그림 2는 통신 설비 구축 및 유지보수 공사 완료 후 진행되는 정산 업무 프로세스를 보여준다. 그림 2에서 보는 것과 같이 K 기업은 작업 내역을 정리한 데이터를 자체적으로 축적하지 않고 원청업체의 클라우드 시스템 상에서만 저장하고 있다. 이로 인해 공사의 특이사항이나 작업 내역 등이 정리된 자체 데이터의 축적이 미비하다는 문제점을 안고 있다.

실제로 축적된 데이터를 조사해 본 결과, 작업 사진 데이터는 남아있지만, 설계된 공사 서류 등은 실물 서류로만 보유하고 있으며, 그 외 작업 내역과 같은 데이터는 모두 원청업체의 클라우드 시스템 상에 저장되고 있음이 확인되었다.

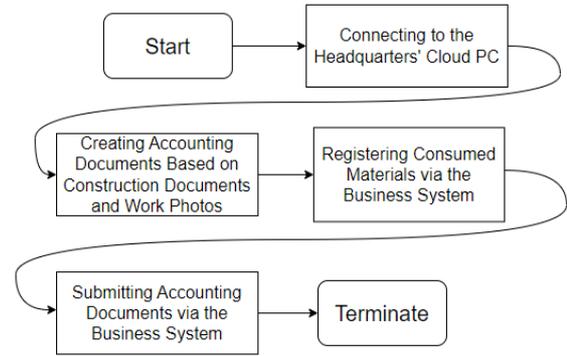


그림 2. 통신 설비 공사 완료 후 정산 업무 프로세스  
Fig. 2. Settlement work process after completion of communication facility

### 3.2 문제점 분석

#### 3.2.1 업무 프로세스

업무 프로세스 관점에서 문제점은 설계자와 작업자 간의 효율적인 커뮤니케이션이 이루어지지 못하고 있다는 것이다. 기존 업무 프로세스에서는 구두 설명, 실물 서류와 실물 서류에 수기로 데이터 기록 등의 비효율적인 수단을 통해 협업이 이루어지고 있다. 이는 실물 서류의 분실 등의 위험성이 있을 뿐 아니라, 설계자가 실물 서류를 통해 설계된 공사에 대해 구두 설명해야 하는 문제가 존재한다. 또한 작업자가 설계자의 구두 설명을 제대로 숙지하지 못하여 전화 등의 연락 수단을 통해 설계자에게 다시 질문하게 되는 경우, 설계자와 작업자의 업무 효율을 떨어뜨리는 원인이 된다.

#### 3.2.2 데이터 축적

데이터 축적 관점에서 문제점은 자체 데이터 축적이 미비하다는 점이다. 기존 업무 프로세스 상에서는 과거 진행했던 공사들의 주요 내역들을 자체적인 데이터로 축적하지 않고 원청업체의 클라우드 PC 상에 데이터를 저장하기 때문에, 작업 내역 등의 데이터를 기업의 데이터 자산으로 활용하기 어렵다.

데이터가 기업의 자산으로 취급되고, 데이터를 정제하고 분석하여 또 다른 가치를 창출해낼 수 있는 가능성이 있으며, 작업 시 발생하는 데이터를 분석함으로써 공사 작업에서 개선할 수 있는 문제점을

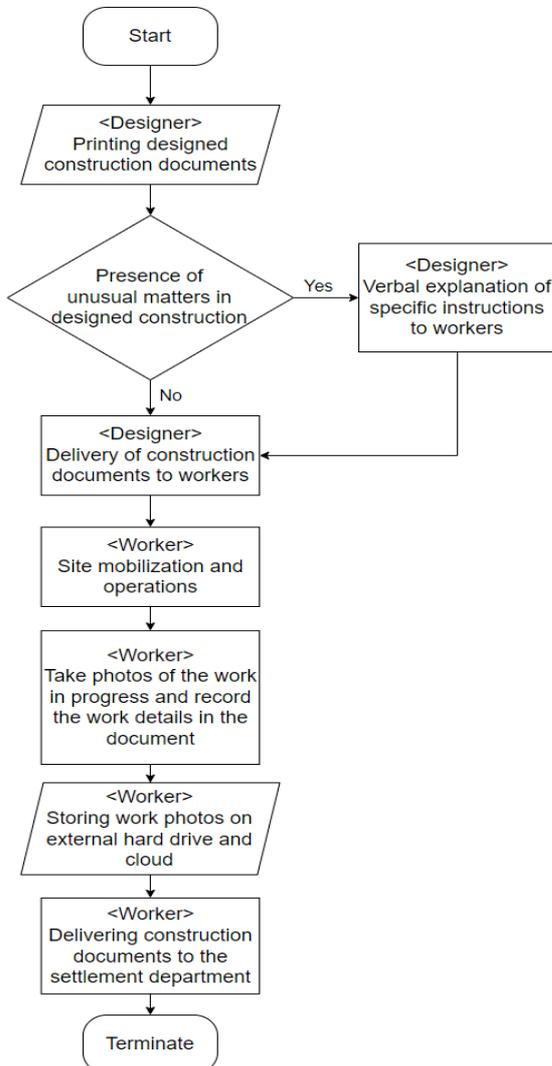


그림 1. 설계자와 작업자 간의 업무 프로세스  
Fig. 1. Work process between designer and worker

발견할 가능성도 있기 때문에 미흡한 자체 데이터 축적은 기업 경쟁력의 약화로 이어질 수 밖에 없다.

### 3.3 개선 방안

앞서 언급한 비효율적인 업무 프로세스와 미비한 데이터 축적을 해결하기 위한 최소 조건을 정리해보면 다음과 같다.

- 비효율적인 업무 프로세스를 개선하기 위해서는 회사에서 내근을 하는 설계자와 현장에서 외근을 하는 작업자 사이에서 좀 더 효율적으로 정확한 데이터를 주고받아야 한다.
- 미비한 데이터 축적을 개선하는 과정에서 프로세스의 복잡성이 증가하는 등, 업무 프로세스 효율성 저하를 야기해서는 안 된다.

또한 내근적인 설계자는 PC를 통해 업무를 수행하는 한편, 외근적인 작업자는 PC를 사용하기 어려운 환경이다. 이러한 환경에서 설계자와 작업자가 가장 원활하게 데이터를 주고받을 수 있는 통신 매체는 모바일 폰이라고 할 수 있다. 그러나 내근적인 설계자의 경우에는 모바일 폰보다 업무용 PC를 이용한 데이터 통신이 더욱 효율적이다. 따라서 설계자와 작업자의 환경을 함께 고려하였을 때, PC와 모바일 두 디바이스 환경 모두에서 동작하는 반응형 웹이 가장 적합하다고 할 수 있다. 모바일 웹에는 다양한 종류의 디바이스로 접속할 수 있는데, 그 디바이스의 화면 사이즈는 서로 다른 경우가 많다. 반응형 웹은 이러한 문제를 해결하기 위해서 접속자의 디바이스 화면 크기에 따라 웹 페이지 구성 요소의 크기 등을 실시간으로 디바이스에 최적화된 뷰를 보여주며[13][14], 이와 대비되는 적응형 웹은 사용자의 특성, 행동, 컨텍스트 등을 고려하여 웹 서비스가 동적으로 조절하여 개인화된 콘텐츠, 인터페이스, 내비게이션 전략을 제공한다[15]. 이에 따라 반응형 웹으로 설계 및 구현한다면 접속 디바이스의 디스플레이 사이즈별로 개발을 진행하지 않아도 되는 장점이 있다. 또한 애플리케이션(앱)으로 설계 및 구현하지 않는 이유는 모바일 환경에서 접속하는 접속자 디바이스의 운영체제가 다를 수도 있기

때문에 디바이스 운영체제별로 개발해야 하는 번거로움이 존재하며, PC 환경에서 업무를 보는 설계자를 위하여 PC용 프로그램 또는 웹을 별도로 개발해야 하기 때문이다.

이러한 개선 방안을 적용하여 설계한 모바일 웹 시스템에서 예상되는 업무 및 데이터 축적 프로세스는 다음과 같다.

‘설계자(Designer)’가 모바일 웹 시스템에 ‘공사 설계(Construction design)’ 데이터를 생성하면 해당 데이터는 자동적으로 데이터베이스에 축적된다. ‘작업자(Worker)’는 모바일 웹 시스템에서 ‘공사 설계’ 데이터를 확인하고, 작업을 수행한다. 작업 완료 후 ‘작업자’는 모바일 웹 시스템에 ‘작업 내역(Work history)’ 데이터를 생성하고, 해당 데이터 또한 자동적으로 데이터베이스에 축적된다.

이를 통해 설계자와 작업자 간의 업무 프로세스, 정보 전달을 개선하며, 업무 프로세스 상에서 데이터를 축적할 수 있다.

그림 3은 업무 프로세스와 데이터 축적의 개선 전후 비교를 나타낸다.

이와 같은 개선 방안을 반영하여 본 논문에서는 통신 설비 기업의 업무 프로세스 개선과 개선된 업무 프로세스상에서의 데이터 축적을 위한 웹 기반 서비스 개발을 위한 시스템을 설계하고자 한다.

## IV. 시스템 설계 및 구성

업무 현황 분석을 통해 도출한 문제점과 개선점을 바탕으로 업무 프로세스 개선 및 데이터 축적을 위한 시스템을 Spring 프레임워크를 기반으로 구현하는 것을 목표로 하여 설계한다. 스프링 프레임워크는 가장 많이 쓰이는 개발 언어 중 하나인 Java를 기반으로 하는 프레임워크이다. 개발 환경 세팅, 프레임워크 내장 함수 활용법 등의 정보가 많기 때문에 접근성이 좋은 언어이며, [16]에서 언급한 것처럼 다른 개발 언어 대비 간단하게 시스템을 구축할 수 있다. 또한 구현 환경을 가정하고 보안 전략 등을 구체적으로 설계하여 개발 전담 부서를 갖추지 못한 기업에서의 시스템 구축, 시스템 골격 코드(Skeleton code) 형성에 기여할 수 있다.

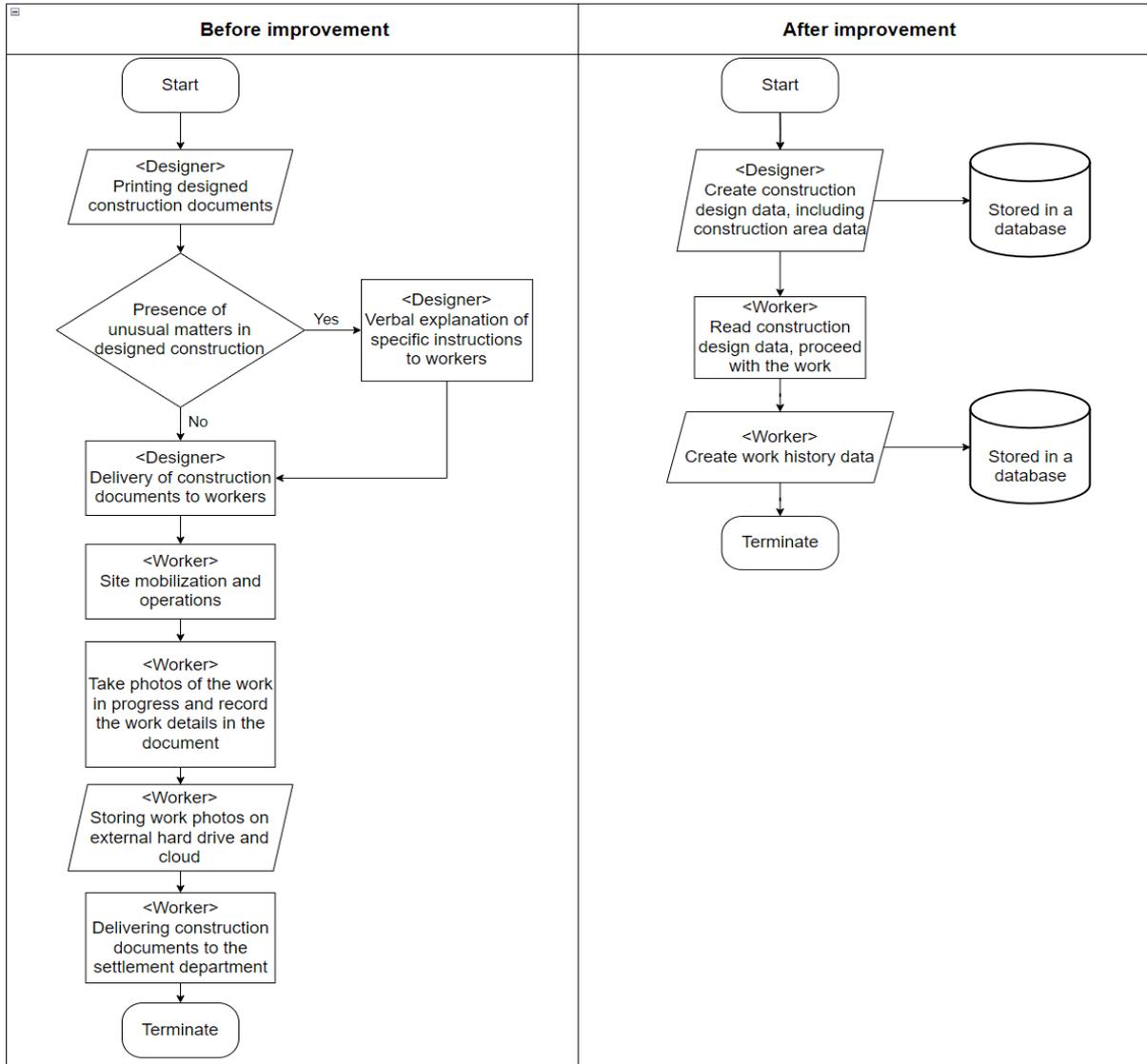


그림 3. 업무 프로세스와 데이터 축적 개선 전후 비교

Fig. 3. Comparison of business process and data accumulation before and after improvement

골격 코드는 시스템 기능의 대부분이 생성되기 전에 구축되는 인프라로, 초기화, 통신, 데이터 공유, 리소스 액세스, 오류 보고, 활동 기록 등을 수행하는 요소들을 포함한다[17]. 즉, 골격 코드는 시스템 구현의 가이드라인 및 뼈대 역할을 하게 된다. 또한 본 연구자가 경험한 바에 의하면 체계적으로 설계를 거치지 않을 경우 시스템 구현 이후 유지보수 단계에서 어려움을 겪을 수 있다. 따라서 체계적인 설계를 위해 UML(Unified Modeling Language)을 바탕으로 설계 모델링을 작성한다. UML은 소프트웨어 개발 시에 일반적으로 사용되며 개발자들이 복잡한 소프트웨어 시스템을 설계하고 관리하는 데 도움을

주는 표준화된 비주얼 모델링 언어이다[18][19][20]. UML 기반 설계를 통해 개발자가 설계한 시스템에 대한 이해도를 향상시키고, 추후 시스템의 유지보수 및 개선 작업에도 도움을 줄 수 있다[21].

#### 4.1 문제 기술서

이 절에서는 업무 프로세스의 문제점을 분석하여 도출한 개선 방안을 바탕으로 하는 시스템을 설계하기 위해 시스템에서 구축되어야 하는 기능을 정리한 문제 기술서를 작성한다.

설계하고자 하는 시스템은 사내 업무 시스템이고, 해당 시스템에 기업의 구성원이 아닌 외부인이 접근할 수 없어야 하므로 사용자 검증 기능을 포함하여 설계한다. 표 1은 통신 설비 업체의 업무 프로세스 개선 따른 문제 기술서를 보여주고 있다. 표 1과 같이 작성된 문제 기술서는 시스템의 도메인 모델링 등 시스템 설계의 기초 자료로 사용한다.

## 4.2 유스케이스 식별 및 명세

### 4.2.1 액터 및 유스케이스 식별

작성한 문제 기술서를 바탕으로 도메인 클래스를 참조하여 시스템의 행위자인 액터(Actor)와 시스템의 유스케이스(Use case)를 도출하고 정의한다.

시스템의 액터는 ‘회원(Member)’, ‘설계자(Designer)’, ‘작업자(Worker)’, ‘관리자(Manager)’로 구분되며 추출한 유스케이스는 표 2에서 요약한 바와 같다. 전체 22개의 유스케이스가 도출되었으며, 각 유스케이스에 식별코드를 부여하였다.

표 2. 유스케이스 식별코드와 유스케이스 명칭

Table 2. Use case identification code and use case name

Identification code	Use case
UC-001	Create account
UC-002	Login
UC-003	Read personal account information
UC-004	Update personal account information
UC-005	Read account list
UC-006	Read account details
UC-007	Delete account
UC-008	Update important account information
UC-009	Create construction design
UC-010	Read construction design list
UC-011	Read construction design details
UC-012	Update construction design
UC-013	Delete construction design
UC-014	Create work history
UC-015	Read work history details
UC-016	Update work history
UC-017	Delete work history
UC-018	Create department
UC-019	Read department list
UC-020	Read department details
UC-021	Update department
UC-022	Delete department

표 1. 문제 기술서

Table 1. Problem statement

Problem statement
<p>Visitors who have not logged in can only access the login function.</p> <p>Members with an account can access the My Page to manage their account information, and member accounts are categorized into three roles: workers, designers, and managers.</p> <p>Because this is a system that deals with internal corporate data, for the sake of company data security, visitors' ability to create accounts should be restricted. Therefore, account creation, deletion, permission, department affiliation, title, and other data that should not be arbitrarily modified by accounts with lower privileges should be managed only by managers. Additionally, managers can create and delete departments and appoint department heads. Each department must have a mandatory department name and department head.</p> <p>Both designers and workers can access the construction design list, construction design details, and work detail view features.</p> <p>Designers can also create, update, and delete construction design data, which consists of multiple construction areas. The construction design data must include a construction number, construction area list, construction type, and designer's name, and special notes can be optionally included.</p> <p>Construction areas included in the design data contain the construction area subscriber's name, subscriber number, department of the workers who will be working on the construction section, and a design map image file.</p> <p>Workers can create, update, and delete detailed work data for each construction area in the design data created by the designers. Work detail data must include the worker's name, construction area, a list of work photos, a list of materials used, and DB measurement values. Special notes during the work process can also be optionally included.</p> <p>Because work is carried out only on designed construction projects, if there is no construction design data, work detail data cannot be entered, and there must be one work detail data for each construction area.</p>

4.2.2 유스케이스 명세

이 절에서는 표 2의 식별된 유스케이스별로 유스케이스 명세를 작성한다.

표 3 및 표 4는 식별코드가 부여된 유스케이스 리스트를 바탕으로 작성한 유스케이스 명세의 예를 보여주고 있으며, 작성한 유스케이스 시나리오의 각 항목에 대해 자세히 살펴보면 다음과 같다.

‘액터’는 해당 유스케이스 명세의 행위자로, ‘액터’가 ‘회원’일 경우 ‘관리자’, ‘설계자’, ‘작업자’로 구분되는 모든 사용자가 해당된다. ‘선행 조건

(Precondition)’은 해당 유스케이스 시나리오에 진입하기 위한 조건이다. ‘기본 흐름(Basic flow)’은 해당 유스케이스 시나리오의 정상적인 시나리오 흐름을 나타내며, ‘선택 흐름(Alternative flow)’은 사용자의 선택에 따라 흐름이 바뀌는 시나리오에서 ‘기본 흐름’과 다른 선택을 했을 경우의 시나리오 흐름이다. ‘예외 흐름(Exception flows)’은 비정상적인 시나리오 흐름을 나타내며 예러 발생 등의 경우가 해당된다. ‘예외 흐름’이 발생할 경우, 시스템은 예러 케이스에 따라 ‘에러 코드(Error code)’를 생성하여 예러에 대한 외 처리를 진행한다.

표 3. ‘공사 설계 상세 조회’ 유스케이스 명세  
Table 3. ‘Detailed inquiry of construction design’ use case specification

Use case	Read construction design details	
Identification code	UC-011	
Actor	Member	
Summary	view construction design details	
Precondition	click on the construction design data to view from the construction design list page	
Post condition	none	
Basic flow	Member	System
		1. access authorization check
		2. view selected construction design data 3. generate construction design detail view page based on the retrieved information
Alternative flows	none	
Exception flows	error code: E3000 error details: authorization verification failed 1. display a message notifying that the requested permission is not granted 2. return to the construction design list view page	

표 4. ‘작업 내역 삭제’ 유스케이스 명세  
Table 4. ‘Delete work history’ use case specification

Use case	Delete work history	
Identification code	UC-017	
Actor	Worker	
Summary	delete work history data	
Precondition	click the delete button on the work history detail view page	
Post condition	none	
Basic flow	Worker	System
		1. access authorization check
	3. select delete confirmation	2. display a delete confirmation alert
		4. delete the work detail data from the database and refresh the construction design detail view page
Alternative flows	3-1) if the worker chooses to cancel the deletion, close the delete confirmation alert and end the work detail data deletion process	
Exception flows	error code: E3000 error details: authorization verification failed 1. display a message notifying that the requested permission is not granted 2. return to the construction design list view page	

### 4.3 유스케이스 다이어그램

이 절에서는 전체 시스템의 유스케이스 다이어그램(Use case diagram)을 작성한다. 그림 4는 식별된 4개의 액터와 22개의 유스케이스 명세를 바탕으로 작성한 유스케이스 다이어그램으로서 세부 내용을 살펴보면 다음과 같다.

‘관리자’와 ‘설계자’, ‘작업자’로 분류되는 ‘회원’은 ‘memberId’와 ‘password’를 입력하여 ‘로그인(login)’할 수 있으며, ‘본인 계정 정보 조회(Read personal information)’ 및 ‘본인 정보 수정(Update personal information)’에 액세스할 수 있다.

‘관리자’는 ‘계정 리스트 조회(Read account list)’ 및 ‘계정 상세 조회(Read account details)’와 ‘계정 생성(Create account)’, 전체 회원 계정의 직책, 부서 등의 ‘계정 중요 데이터 수정(Update account critical information)’, ‘계정 삭제(Delete account)’ 등 전체 회원의 ‘계정(Account)’을 관리할 권한과 ‘계정’이 소속될 ‘부서(Department)’에 대한 관리 권한인 ‘부서 생성(Create department)’, ‘부서 수정(Update department)’ 및 ‘부서 삭제(Delete department)’와 ‘부서 리스트 조회(Read department list)’, ‘부서 상세 조회(Read department details)’ 등의 권한을 갖는다.

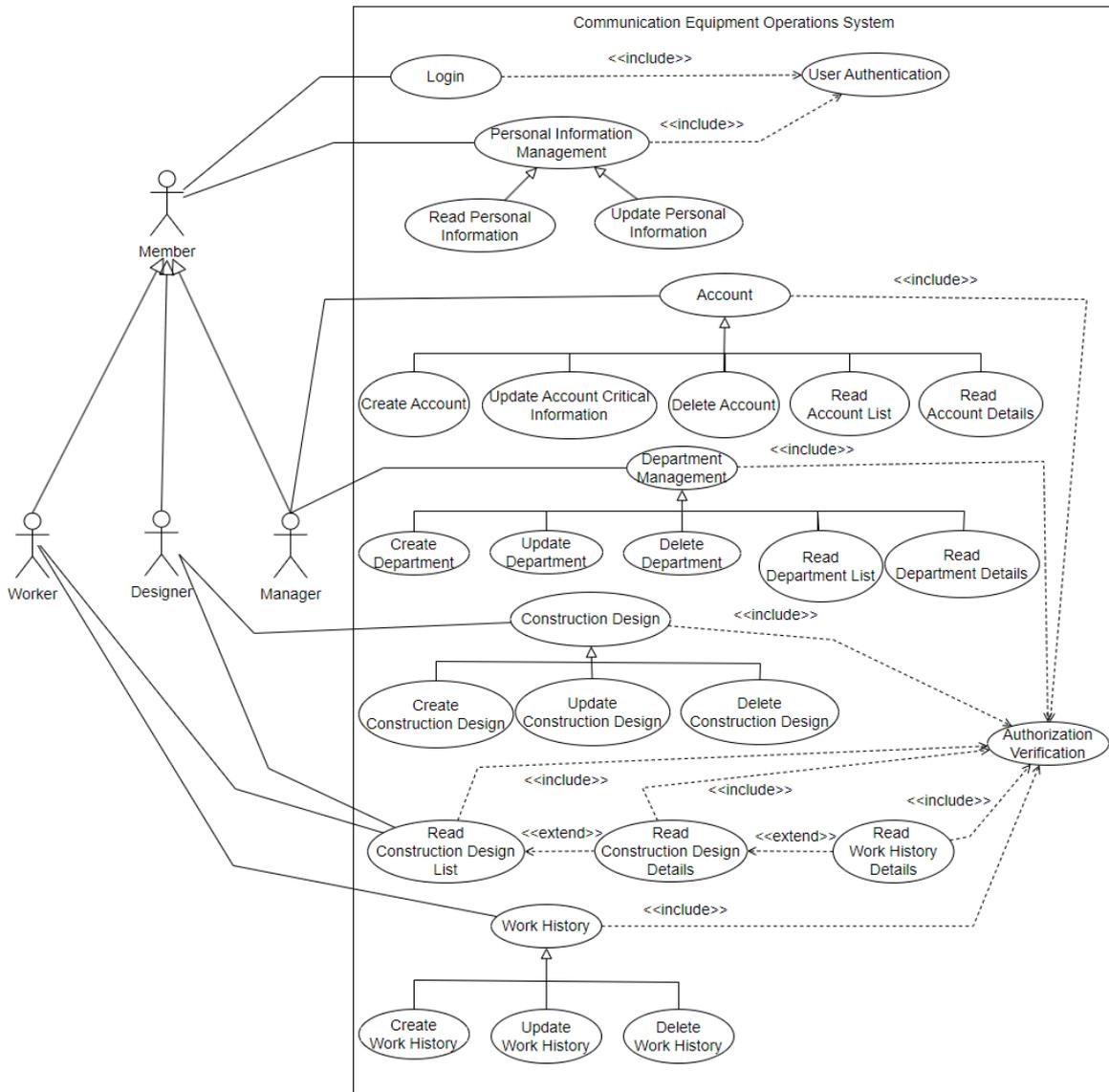


그림 4. 시스템 유스케이스 다이어그램

Fig. 4. System use case diagram

‘설계자’와 ‘작업자’는 공통적으로 ‘공사 설계 리스트 조회(Read construction design list)’ 및 ‘공사 설계 상세 조회(Read construction design details)’, ‘작업 내역 상세 조회(Read work history details)’에 접근할 수 있다. ‘설계자’는 ‘공사 설계 생성(Create construction design)’과 ‘공사 설계 수정(Update construction design)’, ‘공사 설계 삭제>Delete construction design)’의 권한이 주어지며 ‘작업자’는 ‘작업 내역 생성(Create work history)’과 ‘작업 내역 수정(Update work history)’, ‘작업 내역 삭제>Delete work history)’의 권한이 주어진다.

#### 4.4 도메인 모델링

이 절에서는 시스템 설계의 기초가 되는 도메인 모델(Domain model)을 작성한다. 도메인 모델링은 소프트웨어 시스템이 적용될 도메인의 핵심 개념, 구조, 동작을 이해하고 표현하는 과정으로, 도메인 내의 주요 엔티티, 속성, 관계, 제약 조건을 모델링하여 시스템 개발의 방향성과 요구사항을 명확히 하는 데 중요한 역할을 한다[21]. 먼저 도메인 모델링을 위해 표 1의 문제 기술서를 바탕으로 도메인 클래스(Domain class) 후보를 도출한다.

표 5는 1차 도메인 클래스 후보를 도출한 것으로, 문제 기술서에서 구현할 시스템의 키워드가 되는 명사 및 명사구를 기준으로 도출하였다.

표 5의 1차 도메인 클래스 후보에서는 도메인으로 선정하기에 적절하지 않은 요소들이 존재한다.

따라서 1차 도메인 클래스 후보들로부터 부적절한 클래스 후보를 제거하여야 한다.

표 5. 1차 도메인 클래스 후보  
Table 5. Primary domain class candidates

Account	Login	Visitor
Authorization	Member	My Page
Designer	Account management	Manager
Worker	File	Manage personal account
Department	Construction design	Work history
Construction area	Materia	Department management

표 6은 제거한 도메인 클래스 후보와 제거 사유를 보여주고 있다. 표 6에서 보는 것처럼 연산으로 표기하는 것이 바람직한 ‘본인 계정 관리(Manage personal account)’, ‘마이페이지(My page)’, ‘로그인(Login)’, ‘계정 관리(Account management)’, ‘부서 관리(Department management)’를 제거하였고, ‘회원(Member)’이 아닌 ‘방문자(Visitor)’는 시스템의 기능에 접근할 수 없기 때문에 ‘방문자’를 제거하였다. ‘회원’의 속성으로 표기하는 것이 바람직하다고 보이는 ‘권한(Authorization)’ 클래스를 제거하지 않은 이유는 열거형(Enumerate, enum)으로 권한을 별도로 정의하여 추후 권한이 추가될 경우에 관리를 용이하게 하기 위함이다.

표 6. 제거한 도메인 클래스와 제거 사유  
Table 6. Removed domain classes and removal reasons

Remove domain class	Reason for removal
Manage personal account	It is advisable to represent it as an operation of the class
My page	It overlaps with ‘Manage Personal Account’, and it is advisable to represent it as an operation of the class
Login	It is advisable to represent it as an operation of the class
Account management	It is advisable to represent it as an operation of the class
Department management	It is advisable to represent it as an operation of the class
Visitor	Since account creation is conducted with manager privileges and only members can access the system’s features, there is no need for the role of a visitor to represent all users accessing the system

표 7은 1차 도메인 클래스 후보에서 바람직하지 않은 클래스를 제거하고 선정한 2차 도메인 클래스 후보이다.

표 7. 2차 도메인 클래스 후보  
Table 7. Secondary domain class candidates

Account	Authorization	Member
Designer	Manager	Department
Work history	File	Construction area
Material	Construction design	Worker

문제 기술서와 유스케이스 명세를 바탕으로 표 7의 2차 클래스 후보들 간의 연관 관계(Association)와 다중성(Multiplicity)을 표기해보면 그림 5와 같다.

그림 5의 연관 관계 및 다중성에 대해 자세히 살펴보면 다음과 같다.

‘회원’ 클래스는 ‘계정’을 보유한 모든 사용자를 뜻하며, 권한 속성에 따라 ‘설계자’, ‘작업자’, ‘관리자’로 분류되고 ‘회원’ 클래스와 일반화 관계를 갖는다. 또한 한 명의 ‘회원’은 하나의 ‘계정’을 소유하므로 ‘회원’ 클래스와 ‘계정’ 클래스는 일대일 양방향 연관 관계를 갖는다.

또한 ‘계정’은 반드시 하나의 ‘부서’에 소속되지만 ‘부서’에서는 아직 배치된 부서원이 없을 수 있으므로, ‘부서’ 클래스와 ‘계정’ 클래스는 ‘계정’ 클래스가 0일 가능성을 포함한 일대다 양방향 연관 관계를 갖는다.

전체 ‘계정’과 ‘부서’에 대한 관리 권한을 가진 ‘관리자’ 클래스는 ‘계정’과 ‘부서’에 접근할 수 있고, ‘계정’과 ‘부서’에 변경 사항이 있을 경우에 행위자를 알 필요가 없으므로 ‘관리자’ 클래스에서 ‘계정’ 클래스, ‘부서’ 클래스로 단방향 연관 관계를 갖는다.

‘설계자’ 클래스는 ‘공사 설계’의 생성, 수정, 삭제 권한을 갖고 있고, ‘공사 설계’ 클래스에서는 공사 설계 데이터를 작성한 ‘설계자’를 알 필요가 있기 때문에 ‘공사 설계’ 클래스가 0일 가능성을 포함한 일대다 양방향 연관 관계를 갖는다.

‘공사 설계’ 클래스는 1개 이상의 ‘공사 구역’을 갖고, ‘공사 구역’ 클래스는 해당 구역의 공사가 어느 설계 건에 포함되는지 알 필요가 있으므로, ‘공사 설계’ 클래스와 ‘공사 구역’ 클래스는 일대다 양방향 연관 관계를 갖는다.

‘작업자’ 클래스는 ‘작업 내역’의 생성, 수정, 삭제 권한을 갖고 있고, ‘작업 내역’ 클래스는 작업 내역 데이터를 작성한 ‘작업자’를 알 필요가 있기 때문에 ‘작업 내역’ 클래스가 0일 가능성을 포함한 일대다 양방향 연관 관계를 갖는다.

또한 ‘작업 내역’ 클래스는 ‘공사 구역’ 클래스와 ‘작업 내역’ 클래스가 0일 가능성을 포함한 일대일 양방향 연관 관계를 갖는데, 한 건의 ‘공사 설계’는 한 건 이상의 ‘공사 구역’ 데이터를 갖게 되고, ‘작업자’가 공사를 진행한 이후에는 한 건의 ‘공사 구역’마다 한 건의 ‘작업 내역’을 작성해야 하기 때문이다.

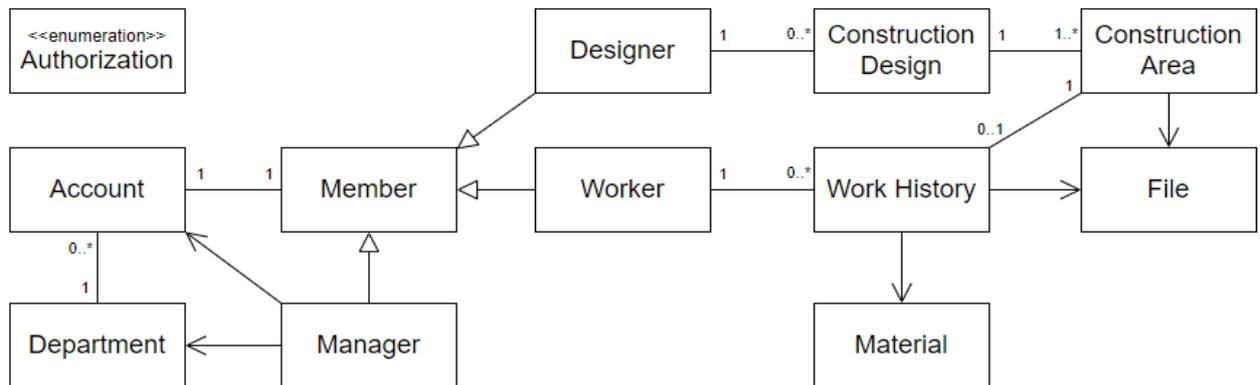


그림 5. 도메인 클래스들 간의 연관 관계 및 다중성  
Fig. 5. Association relationships and multiplicity among domain classes

‘자재’ 클래스는 작업에 사용되는 자재의 재고 수량 등을 관리하기 위한 클래스로, 작업 시 사용 수량을 입력하게 된다. 따라서 ‘작업 내역’ 클래스와 단방향 연관 관계를 갖는다.

‘파일’ 클래스는 ‘작업 내역’과 ‘공사 구역’의 사진 등의 파일을 효율적으로 관리하기 위한 클래스로, ‘작업 내역’ 클래스 및 ‘공사 구역’ 클래스와 단방향 연관 관계를 갖는다. 그림 6은 도메인 클래스들을 대상으로 속성(Attributes) 및 연산(Operations)을 도출한 클래스 다이어그램(Class diagram)이다.

### 4.5 시퀀스 다이어그램

이 절에서는 도출된 그림 6의 클래스 다이어그램의 연산과 유스케이스 명세를 바탕으로 전체 시스템 기능을 시퀀스 다이어그램(Sequence diagram)으로 작성한다. 시퀀스 다이어그램은 앞서 작성한 유스케이스 명세의 각 유스케이스가 시스템의 각 기능들이 어떠한 흐름을 통해 동작하는지 나타낸다.

그림 7은 ‘설계자’의 ‘공사 설계 생성(Create construction design)’ 시퀀스 다이어그램이다.

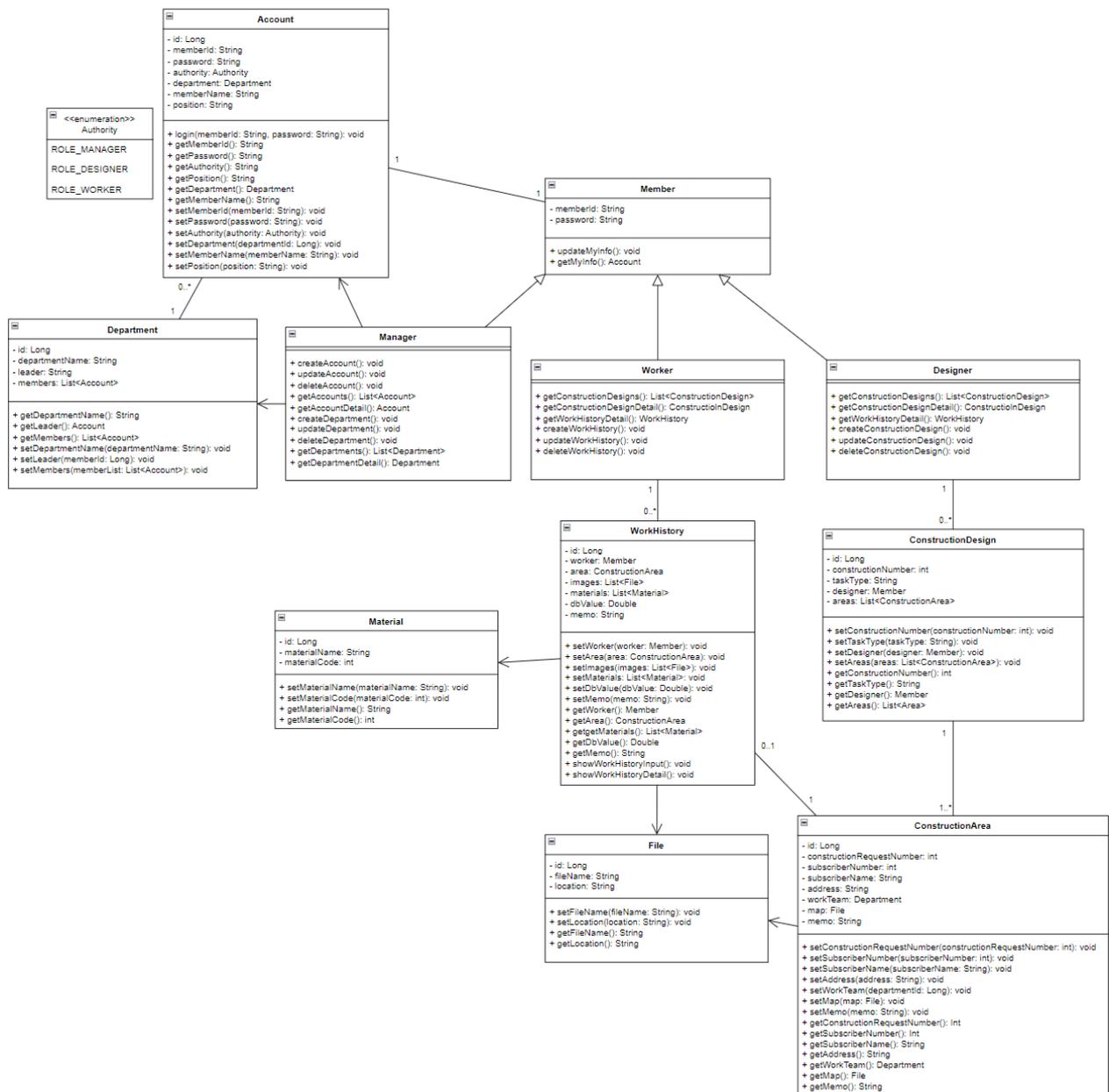


그림 6. 도메인 클래스 속성 및 연산 도출  
Fig. 6. Deriving domain class attributes and operations

‘설계자’가 공사 설계 데이터 생성 요청 메시지를 전송하면 ‘공사 설계(ConstructionDesign)’는 ‘설계자’에게 공사 설계 데이터 입력 페이지를 보여준다. ‘설계자’는 해당 페이지에 공사 설계 데이터를 입력하여 ‘공사 설계’에 전송한다. 이후 ‘공사 설계’는 ‘설계자’가 입력한 데이터 중 구역별 데이터를 ‘공사 구역(ConstructionArea)’에 전송한다. 또한 ‘공사 구역’은 ‘설계자’가 전송한 데이터들 중 사진 등의 파일 데이터를 ‘파일(File)’에 전송한다. ‘파일’은 파일 저장 성공 여부를 ‘공사 구역’에 반환하고, ‘공사 구역’은 공사 구역 데이터 생성 성공 여부를 ‘공사 설계’에 반환한다. 이후 ‘공사 설계’는 ‘설계자’에게 공사 설계 생성 성공 여부를 반환한다.

그림 8은 ‘작업자’의 ‘작업 내역 수정(Update work history)’ 시퀀스 다이어그램이다. 작업 내역 데이터 삭제 시퀀스와 마찬가지로 ‘공사 설계’를 통해 공사 설계 데이터를 불러오는 것부터 시작한다. ‘작업자’가 수정할 작업 내역 데이터의 입력을 완료하면 입력된 데이터를 ‘작업 내역’에 전송한다. ‘작업 내역’은 입력된 데이터 중 파일 데이터를 ‘파일’에 전송하고, ‘파일’은 ‘작업 내역’에 파일 저장 성공 여부를 반환한다. 다음으로 ‘작업 내역’은 ‘자재(Material)’에 자재 데이터 저장 메시지를 전송하고

‘자재’는 자재 데이터 저장 성공 여부를 반환한다. 이후 ‘작업 내역’은 작업 내역 데이터 수정 성공 여부를 반환한다.

#### 4.6 보안 및 권한 관리

시스템의 설계 및 구현을 위한 다이어그램을 작성하였으나, 아직 시스템 액세스 권한 관리나 시스템의 보안 수단에 대한 설계는 이루어지지 않았다. 본 논문에서는 이 두 가지 문제점을 동시에 해결할 수 있는 ‘JWT(JSON Web Token)’를 보안 수단으로 채택하였다. ‘JWT’의 정의에 대해 살펴보면, [22]에서는 ‘JWT’는 두 당사자 간에 전송될 클레임(Claim)을 나타내는 소형 URL 보안 수단이라고 정의하였다.

‘JWT’는 일반적으로 두 종류의 토큰을 하나의 세트로 구성하는데, 인증 수단으로 사용되는 ‘Access Token’과 ‘Access Token’이 만료되었을 경우 재발급(Refresh)을 위해 사용되는 ‘Refresh Token’으로 구성된다. 토큰에는 토큰 사용자를 식별할 때 사용되는 ‘식별자’, ‘토큰 만료 시간’, ‘권한 정보’ 등의 데이터를 포함할 수 있다.

설계한 시스템에서 ‘JWT’를 발급하는 과정은 그림 9와 같다.

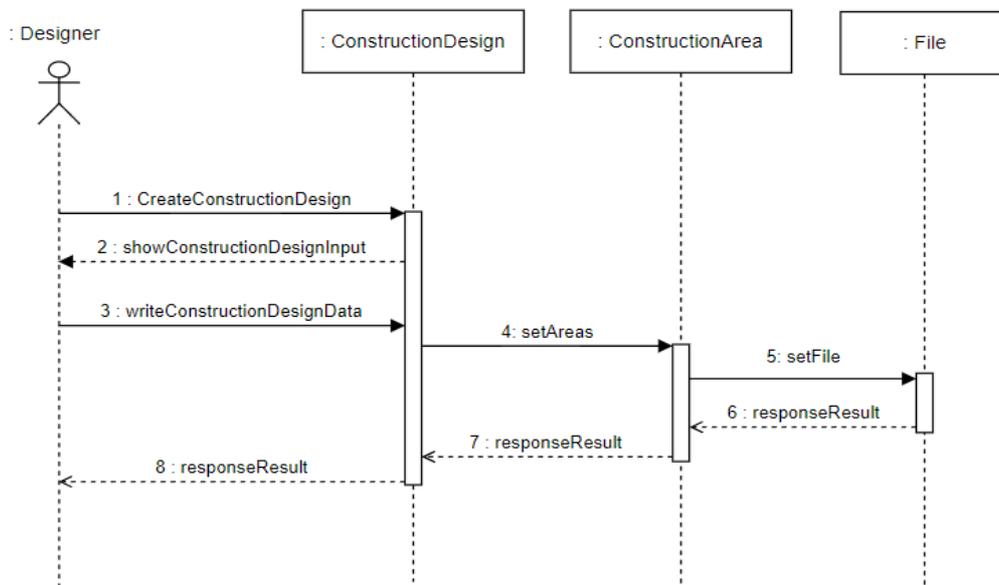


그림 7. ‘공사 설계 생성’ 시퀀스 다이어그램  
 Fig. 7. ‘Create construction design’ sequence diagram

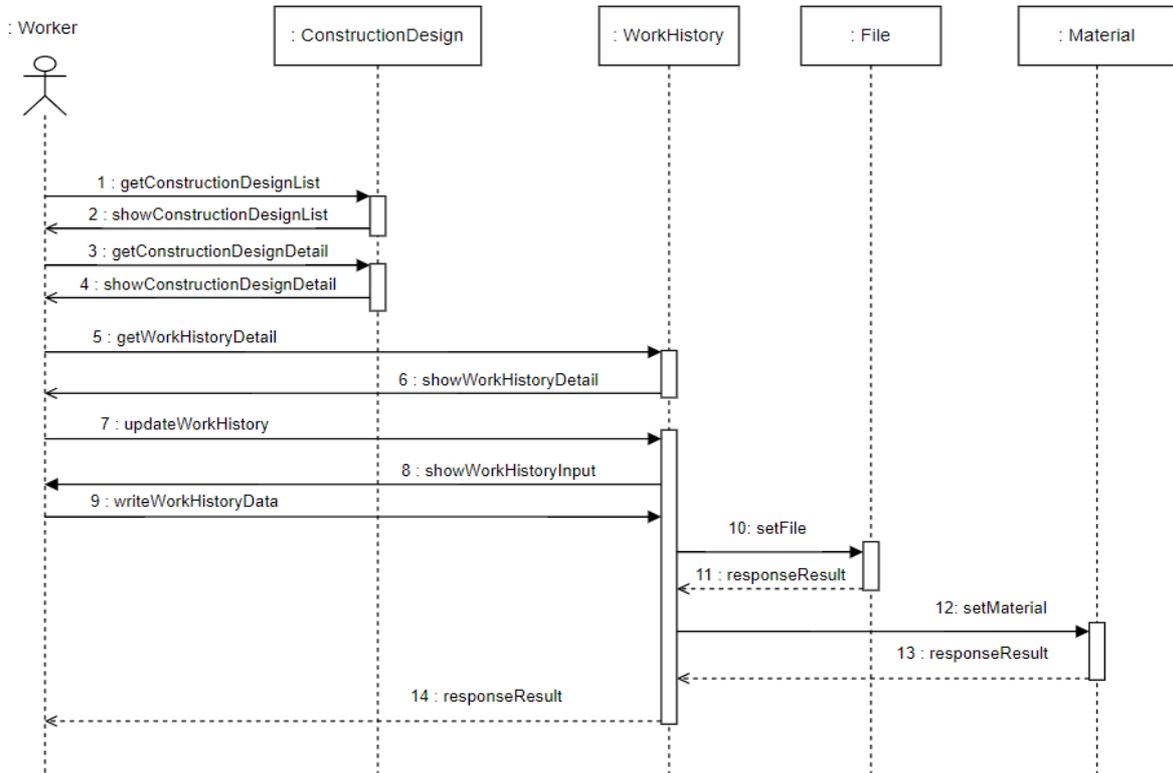


그림 8. '작업 내역 수정' 시퀀스 다이어그램  
 Fig. 8. 'Update work history' sequence diagram

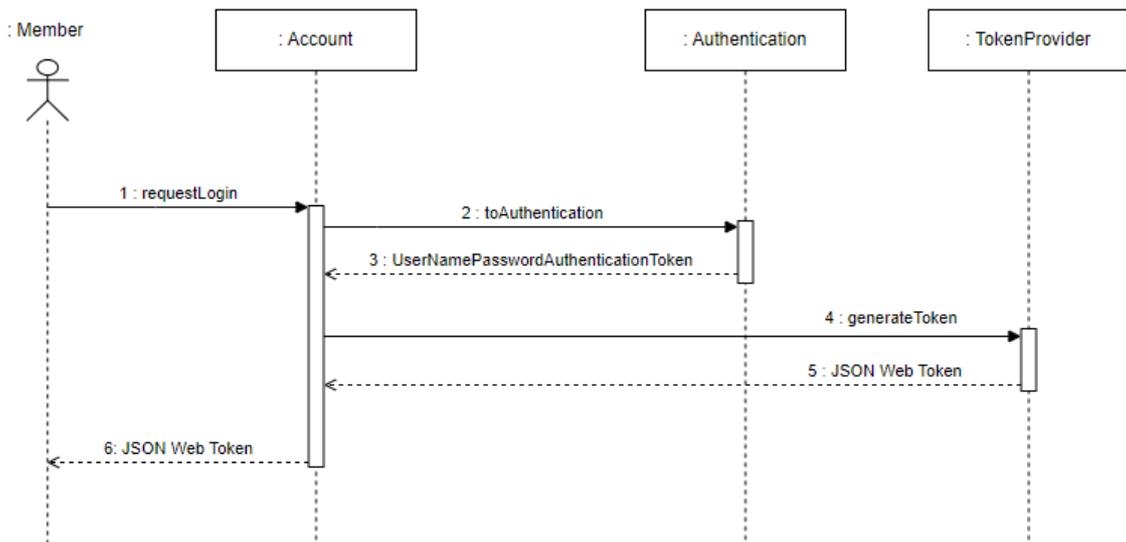


그림 9. 로그인 시 '토큰 발급 과정'을 나타낸 시퀀스 다이어그램  
 Fig. 9. Sequence diagram depicting 'token issuance process' during login

그림 9는 앞서 설계한 '로그인(Login)' 시퀀스 다이어그램에서 토큰 발급 과정을 나타낸 것으로, 스프링 프레임워크를 기준으로 작성하였다. 그림 9의 시퀀스 다이어그램에서 'Authentication'은 인증, 권

한 등을 처리하는 스프링 프레임워크의 하위 프레임워크인 Spring security의 클래스이며, 'TokenProvider'는 토큰 검증 등의 토큰 관리와 관련된 로직을 가진 커스텀 클래스이다.

‘회원’이 ‘memberId’와 ‘password’를 입력하여 로그인 요청하면 ‘계정’에서 해당 정보를 ‘Authentication’ 클래스로 전송하여 ‘UserNamePasswordAuthenticationToken’을 생성한다. ‘UserNamePasswordAuthenticationToken’은 스프링 시큐리티의 사용자 인증에 사용되는 간단한 인증 토큰으로, ‘사용자 ID’, ‘인증 정보’(주로 password), ‘사용자 권한 정보’, ‘사용자 인증 상태’로 구성되어 있다. 생성된 ‘UserNamePasswordAuthenticationToken’을 ‘JWT’의 생성, 토큰 복호화 등 토큰을 관리하는 ‘TokenProvider’ 클래스로 전송하면, ‘TokenProvider’에서 만료 기한 정보를 추가하고 암호화한 뒤, ‘Access Token’과 ‘Refresh Token’을 생성한다. 이후 이를 ‘계정’으로 전송하고, ‘계정’은 ‘회원’에게 토큰을 발급한다. 또한 설계한 시스템에서 사용자가 시스템의 기능에 대한 접근을 요청하면 시스템이 요청자의 인증 정보를 저장하는 과정은 그림 10과 같다.

그림 10을 자세히 살펴보면, ‘회원’이 시스템의 기능에 대한 접근을 요청할 때, 로그인을 통해 발급 받은 토큰을 헤더(header) 부분에 담아서 요청과 함께 전송한다. ‘회원’의 요청을 받은 시스템은 가장

먼저 ‘JwtFilter’를 통해 필터링을 수행한다. ‘JwtFilter’는 스프링 프레임워크의 웹 필터를 상속받는 커스텀 클래스로, 요청에 대한 필터링을 수행하는 클래스이다. ‘JwtFilter’는 사용자 요청의 헤더 부분에서 토큰을 꺼내고(resolveToken), 해당 토큰을 ‘TokenProvider’로 보내 토큰의 유효성과 만료 기한 등의 정보에 대한 검증을 진행한다(validateToken). 이후 ‘JwtFilter’는 ‘TokenProvider’로부터 토큰 검증 결과를 반환받고, 다시 ‘TokenProvider’로 인증 정보를 요청한다(getAuthentication). ‘TokenProvider’는 토큰을 복호화 하고, 토큰 속의 인증 정보를 ‘UserDetailsService’로 전송한다. ‘UserDetailService’는 토큰의 인증 정보를 데이터베이스와 대조하여 계정 정보를 찾고, 해당 데이터를 ‘UserDetails’의 형태로 ‘TokenProvider’에 반환한다. ‘TokenProvider’는 ‘UserDetailsService’로부터 받은 ‘UserDetails’ 데이터를 ‘UserNamePasswordAuthenticationToken’의 형태로 ‘JwtFilter’에 반환하며, ‘JwtFilter’는 ‘SecurityContext’에 저장한다(setAuthentication). ‘SecurityContext’는 사용자 인증 정보가 필요한 상황에 해당 정보를 꺼낼 수 있도록 하는 저장소이다.

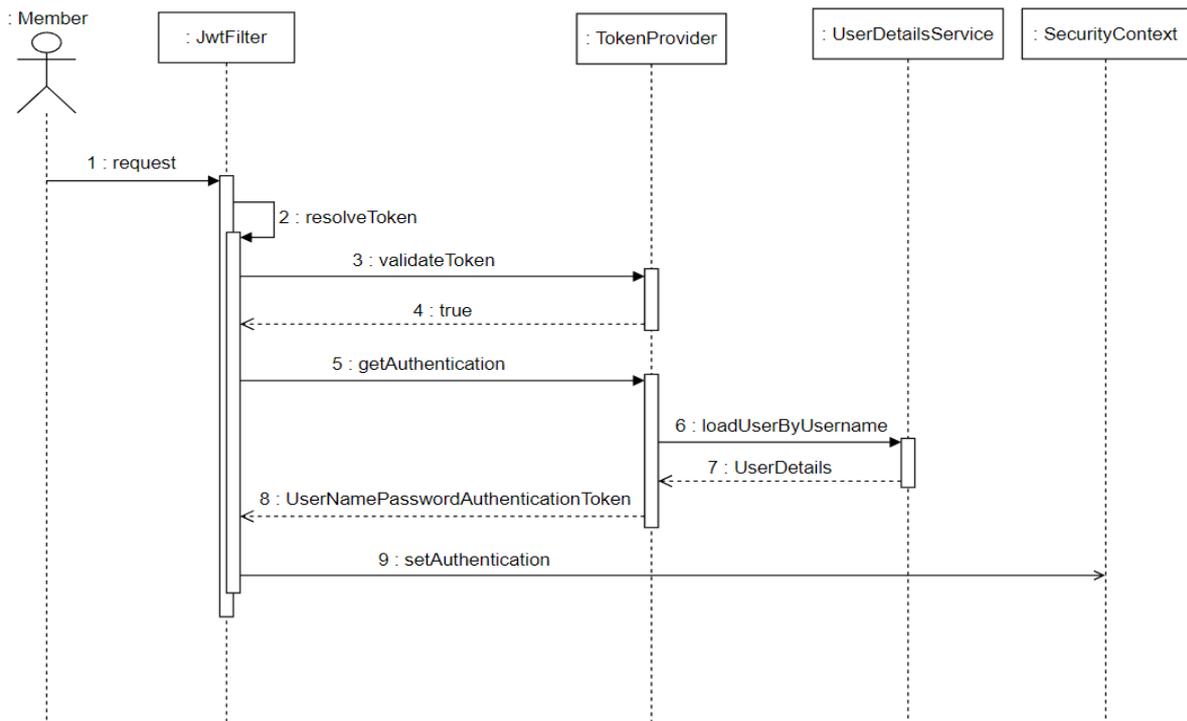


그림 10. '사용자 인증 정보 저장' 시퀀스 다이어그램  
 Fig. 10. 'User authentication information storage' sequence diagram

사용자 인증 정보에는 ‘회원’의 권한 정보도 포함되어 있기 때문에, ‘회원’이 사용자 인증 또는 권한 검증을 수행해야 하는 기능에 접근하면 시스템은 ‘SecurityContext’에 저장된 사용자 인증 정보를 통해 사용자 인증을 수행하거나, 사용자 인증 정보 속의 권한 정보를 활용하여 사용자 권한 검증을 수행하게 된다.

#### 4.7 데이터베이스 구축을 위한 E-R 다이어그램

이 절에서는 문제 기술서와 앞서 설계한 내용을 바탕으로 데이터베이스를 구축하기 위하여 엔티티(Entity)와 엔티티 간의 관계(Relationship)를 설계하여 E-R 다이어그램(Entity relationship diagram)을 작성한다. E-R 다이어그램은 데이터베이스 구축에 필요한 엔티티와 엔티티의 속성과 속성의 타입, 그리고 엔티티들의 관계를 나타낸 다이어그램이다. E-R 다이어그램을 활용한 데이터베이스 설계는 데이터의 구조와 관계를 명확하게 시각화하여 시스템 내의 데이터를 이해하는 데 도움을 준다. 이는 시스템을 이용하며 축적된 데이터를 활용하기 위한 사전

작업인 데이터 정제 작업에 기여할 수 있다. 그림 11은 클래스 다이어그램의 속성과 연관 관계를 바탕으로 엔티티의 속성과 엔티티 간의 관계를 도출한 E-R 다이어그램이다.

그림 11의 E-R 다이어그램을 자세히 살펴보면, 엔티티의 속성에서 ‘PK’는 엔티티의 식별자로, 기본 키(Primary key)를 나타낸다. 기본 키를 통해 엔티티를 식별할 수 있어야 하므로 기본 키는 고유성을 갖는 특성이 있다. ‘FK’는 연관 관계를 가지는 다른 엔티티의 기본 키를 나타내는 외래 키(Foreign key)이다. ‘U’는 해당 속성의 고유성(Unique)을 나타내며, 중복된 값을 가질 수 없다. 또한 엔티티 속성의 ‘NOTNULL’은 빈 값(Null)을 가질 수 없다는 뜻으로, 해당 속성이 필수 입력값임을 나타낸다.

#### 4.8 목표 시스템의 구성도

이 절에서는 설계한 내용을 바탕으로 목표 시스템의 구성도 및 아키텍처를 제시한다. 이를 통해 설계한 시스템의 구성을 명확히 하여 앞서 UML을 기반으로 설계한 내용과 함께 골격 코드 생성에 기여한다.

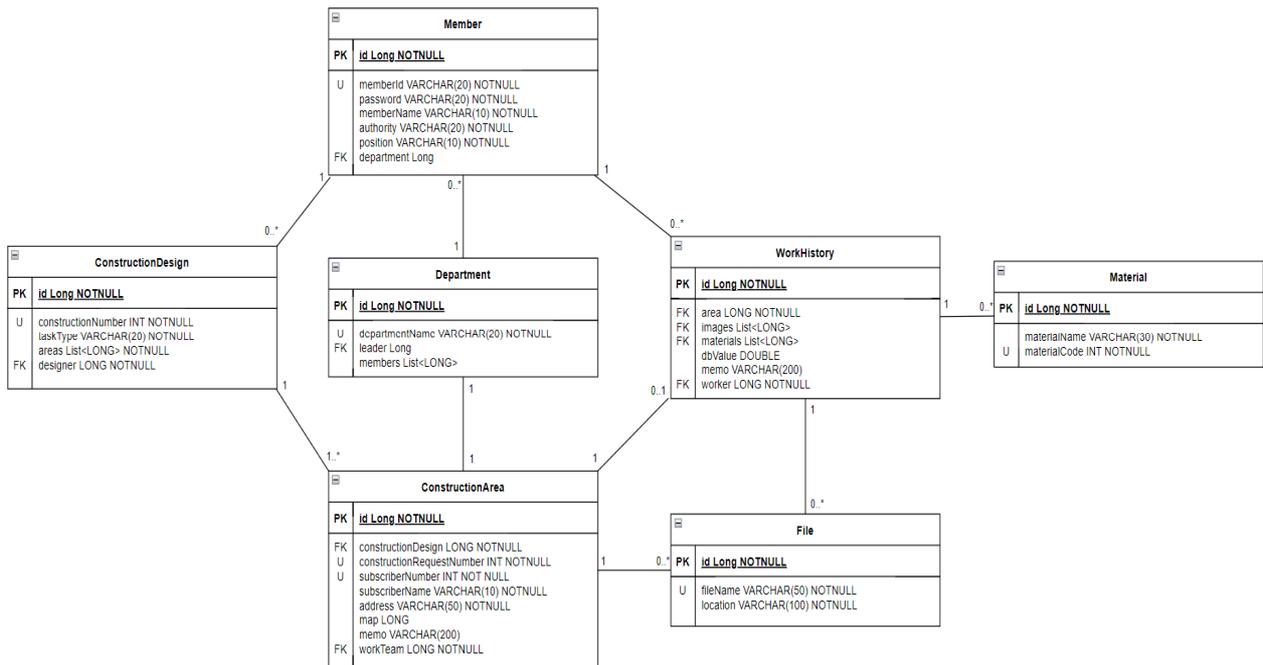


그림 11. 데이터베이스 구축을 위한 E-R 다이어그램  
Fig. 11. Entity relationship diagram for database construction

#### 4.8.1 시스템 구성도

본 논문에서 연구 표본으로 선정한 기업과 같은 통신 설비 시공 업체는 자체적으로 서버 시스템을 갖추기 어려운 환경이다. 따라서 서버 시스템을 클라우드 컴퓨팅 형태로 제공하는 AWS(Amazon Web Service)를 채용하는 것을 기준으로 시스템 구성도를 설계한다. 그림 12는 본 논문에서 설계한 내용을 바탕으로 작성한 시스템 구성도를 나타낸다.

그림 12의 시스템 구성도를 자세히 살펴보면 다음과 같다. 사용자의 컴퓨터 시스템을 뜻하는 ‘클라이언트(Client)’는 ‘PC 클라이언트’와 ‘모바일 클라이언트’의 두 형태로 나눈다. ‘PC 클라이언트’는 앞서 분류했던 액터 중 내근적인 ‘관리자’와 ‘설계자’이며, ‘모바일 클라이언트’는 외근적인 ‘작업자’가 해당된다. ‘클라이언트’는 디바이스의 ‘브라우저(Browser)’를 통해 ‘웹 서버(Web server)’에 콘텐츠를 ‘요청(Request)’한다. ‘웹 서버’는 ‘클라이언트’의 요청 중 연산이나 ‘데이터베이스(Database)’에 접근이 필요한 경우, ‘웹 응용 서버(Web application server)’에 요청된 내용을 전달한다. ‘웹 응용 서버’는 요청된 연산을 수행하고 ‘데이터베이스’와 ‘쿼리(Query)’문을 통해 소통하며, 결과를 ‘웹 서버’에 반환한다. 이후 ‘웹 서버’는 요청에 대한 결과를 ‘클라이언트’에게 ‘응답(Response)’한다.

#### 4.8.2 소프트웨어 아키텍처

시스템 구성도를 통해 시스템 구조를 설계하였

으나, 시스템의 기능과 내부 로직이 어떠한 흐름을 통해 작동되는지 등의 소프트웨어 측면에서의 설계는 미흡하다. 따라서 이 절에서는 ‘클라이언트’가 시스템의 기능을 사용할 때 시스템에서 어떠한 과정을 통해 해당 기능을 서비스하는지를 나타내는 소프트웨어 아키텍처를 설계한다. 본 논문에서 설계한 시스템은 스프링 프레임워크를 기반으로 설계하였기 때문에 스프링 프레임워크 기반 웹 시스템 아키텍처를 작성한다.

그림 13은 본 논문에서 설계한 시스템을 바탕으로 작성한 소프트웨어 아키텍처이다. 그림 13의 구성을 자세히 살펴보면, ‘컨트롤러(Controller)’는 ‘클라이언트’의 ‘요청’을 처리하기 위한 함수를 ‘서비스(Service)’로부터 호출하고, ‘뷰(View)’에게 ‘요청’에 대한 처리 결과를 전송하는 역할을 수행한다. ‘서비스’는 시스템의 각 기능을 처리하는 비즈니스 로직(Business logic)을 함수 형태로 소유한다. 또한 ‘서비스’는 비즈니스 로직 수행 중 ‘데이터베이스’에 접근이 필요한 상황에서 ‘레포지토리(Repository)’의 관련 DB 로직(Logic)을 호출하거나 비즈니스 로직 수행 결과를 전송한다. ‘레포지토리’는 ‘데이터베이스’에 접근하기 위한 로직을 가지며 ‘데이터베이스’를 관리하고, ‘뷰’는 처리된 ‘요청’의 결과 데이터를 바탕으로 화면을 재구성한다. 또한 ‘컨트롤러’, ‘뷰’, ‘서비스’는 ‘DTO(Data Transfer Object)’라는 읽기 및 수정이 가능한 객체 형태로 데이터를 주고받는다. 또한 설계한 소프트웨어 아키텍처를 통해 시스템 기능의 흐름을 살펴보면 다음과 같다. 시스템의 ‘사용자(Users)’는 ‘브라우저’를 통해 시스템과 소통한다.

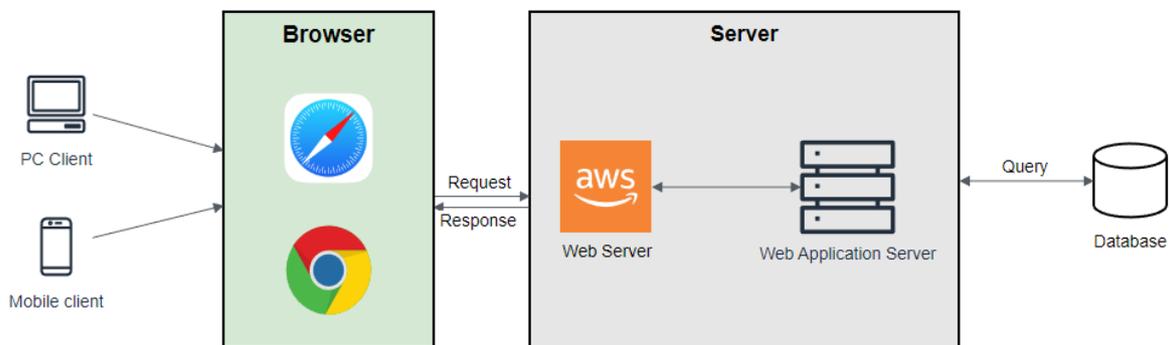


그림 12. 시스템 구성도  
Fig. 12. System architecture diagram

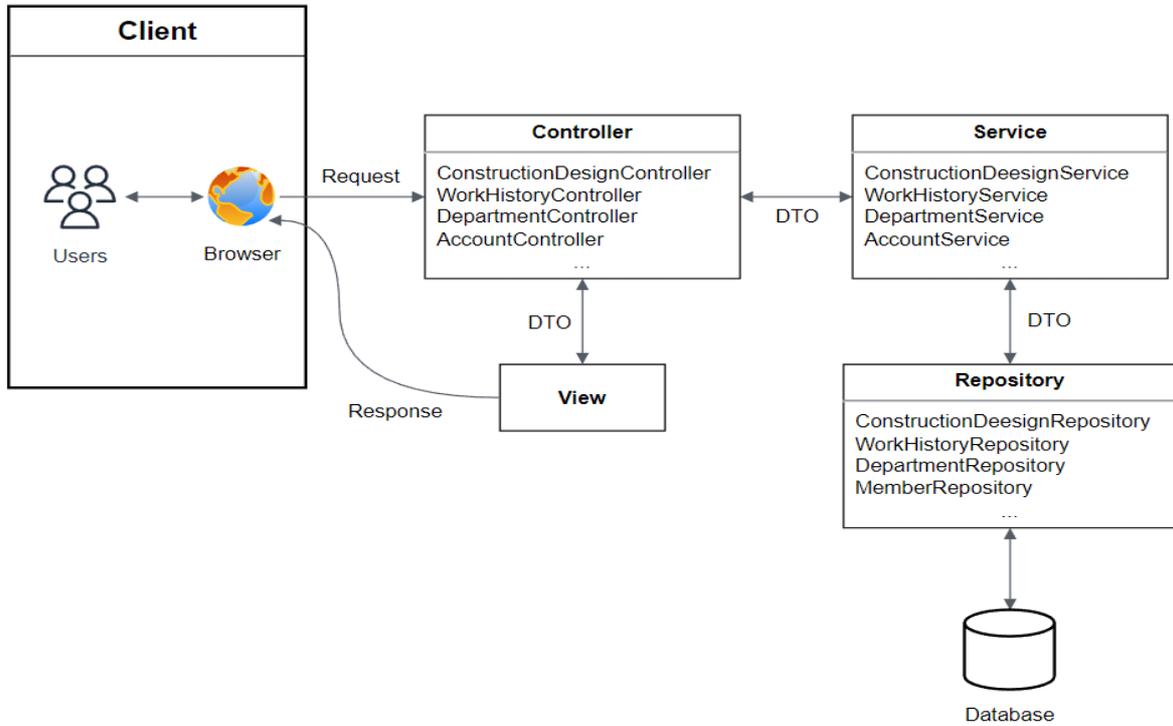


그림 13. 소프트웨어 아키텍처  
Fig. 13. Software architecture

‘사용자’가 ‘브라우저’를 통해 시스템에 콘텐츠를 ‘요청’하면 ‘컨트롤러’는 해당 ‘요청’의 처리를 위해 필요한 데이터를 ‘DTO’에 담아 ‘서비스’로 전송하고, 비즈니스 로직을 ‘서비스’로부터 호출한다. ‘서비스’는 ‘컨트롤러’로부터 받은 데이터와 호출된 비즈니스 로직을 통해 ‘요청’을 처리하며, 이 과정에서 ‘데이터베이스’에 접근해야 할 필요가 있는 로직을 ‘레포지토리’에서 호출하거나 ‘요청’을 처리한 결과 데이터를 ‘DTO’에 담아 전송한다. ‘레포지토리’는 ‘서비스’가 호출한 데이터베이스 관련 로직을 처리하고 결과를 ‘DTO’에 담아 반환하거나, ‘요청’에 대한 결과 데이터를 ‘서비스’로부터 전달받아 이를 ‘데이터베이스’에 저장한다. 이후 ‘서비스’는 ‘사용자’의 ‘요청’에 대한 처리 결과와 필요한 데이터를 ‘DTO’에 담아 ‘컨트롤러’에 전송하고, ‘컨트롤러’는 해당 데이터를 ‘뷰’에 전송하며, ‘뷰’는 처리 결과와 데이터를 바탕으로 ‘사용자’의 화면을 업데이트한다.

## V. 결 론

본 논문에서는 통신 설비 시공 업체의 업무 프로

세스를 개선하고 자체 데이터를 축적하기 위한 모바일 웹 기반 시스템을 설계하였다. 설계한 모바일 웹 기반 시스템을 사용함으로써 프로세스 단계를 축소하여 업무 효율을 향상시키고, 설계자와 작업자 간의 효율적인 정보 전달이 가능하다.

또한 설계자가 공사 설계 데이터를 시스템에 업로드하거나, 작업자가 작업 내역 데이터를 업로드할 때 자동적으로 데이터베이스에 해당 데이터를 저장하여 업무 프로세스를 방해하지 않고 데이터를 축적할 수 있다.

설계한 모바일 웹 기반 시스템은 반응형 모바일 웹을 채택하여 사용자의 디바이스 및 디바이스의 운영체제에 제한되지 않는 시스템을 구축하였다. 시스템의 데이터베이스는 실제 업무에서 사용되는 업무 용어들을 바탕으로 설계하였으며, 업무 수행 시스템을 활용함으로써 자동적으로 유의미한 데이터를 축적할 수 있는 기능을 갖추었다.

해당 축적 데이터를 활용하여 업무 개선을 위한 분석을 진행하거나, 기업의 자체 축적 데이터로 활용하여 데이터 산업에 진출하는 등 기업의 자생력 향상에 기여하여 의존성을 낮추고 기업 경쟁력 향상에 도움이 될 수 있다.

목표한 모바일 웹 기반 시스템은 UML을 사용하여 객체지향 설계하였다. 표본으로 선정한 기업의 현행 업무 프로세스를 분석하여 문제점과 개선점을 파악해 문제 기술서를 작성하였으며, 해당 문제 기술서를 바탕으로 시스템이 갖춰야 할 기능 등의 요소를 도출하여 유스케이스 다이어그램을 작성하였다. 작성한 유스케이스와 문제기술서를 바탕으로 도메인 클래스를 도출하였으며, 도출된 도메인 클래스 간의 연관 관계와 다중성을 표기하고 각 클래스의 속성과 연산을 도출하여 클래스 다이어그램을 작성하였다. 각 연산의 흐름을 나타내기 위해 클래스 다이어그램과 유스케이스를 바탕으로 시퀀스 다이어그램을 작성하였으며, 구현 단계에서 개발자의 이해를 돕기 위해 보안 토큰 발급 및 권한 검증 단계의 흐름을 시퀀스 다이어그램으로 작성하였다. 또한 축적한 데이터의 효율적인 관리를 위해 데이터베이스 E-R 다이어그램을 설계하였다. 이와 같은 객체지향 설계는 목표한 모바일 웹의 구현 단계에서 골격 코드의 생성과 시스템 유지보수가 용이하다는 장점이 있다.

그러나 본 논문에서 설계한 시스템은 원청업체의 업무 시스템상에서 행해지는 업무를 제외한 공사 설계자와 공사 담당자 사이에 이루어지는 프로세스만을 개선한 것이며, 원청업체의 업무 시스템과 연계하여 전체 업무 프로세스를 개선할 여지가 남아 있다. 또한 해당 시스템을 통해 축적한 데이터는 이름, 주소 등의 개인 정보 데이터를 포함하고 있기 때문에 프라이버시 보호를 위해 누구의 데이터인지 식별 불가 데이터로 가공하는 익명화가 필요하다 [23]. 또한 축적된 데이터를 활용하는 방안도 과제로 남아있다. 따라서 향후에는 개인 정보 데이터 익명화 방안을 모색하여 고객의 프라이버시를 존중하며, 고객의 데이터를 보호하는 데이터 활용 모델을 개발하는 등의 연구를 통해 시스템을 개선해 나가야 할 것이다.

## References

- [1] V. Mayer-Schönberger and K. Cukier, "Big Data: A revolution that will transform how we live, work, and think", Houghton Mifflin Harcourt, Mar. 2013.
- [2] S. B. Choi, "Media Management", Communication Books, Feb. 2013.
- [3] S. W. Kim, "A study about the influence of the Big Data on the competitiveness of enterprises", Master's Degree Thesis at Kyunggi University Graduate School, 2014.
- [4] J. W. Shin, H. G. Ryu, and D. R. Lee, "Information-centered Design Work Process for Effective Design Management", Journal of the Architectural Institute of Korea Structure & Construction, Vol. 24, No. 4, pp. 133-141, Apr. 2008.
- [5] Y. Feng and S. D. Kwon, "A study on the Effect of Process, IT, and Organization Characteristics on Business Process Virtualizability", Information Systems Review, Vol. 24, No. 4, p.119-142, Nov. 2022.
- [6] T. Fehrer, D. A. Fischer, S. J. J. Leemans, M. Röglinger, and M. T. Wynn, "An assisted approach to business process redesign", Decision Support Systems, Vol. 156, No. 113749, May 2022. <https://doi.org/10.1016/j.dss.2022.113749>.
- [7] A. Kumar and R. Liu, "Business Workflow Optimization Through Process Model Redesign", IEEE Transactions on Engineering Management, Vol. 69, No. 6, pp. 3068-3084, Dec. 2022. <https://doi.org/10.1109/TEM.2020.3028040>.
- [8] T. P. Tsiri, I. A. Daniyan, and K. Mpfu, "Development of a Business Process Modelling Framework for Continuous Improvements in Organisations", 2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 675-680, Dec. 2022. <https://doi.org/10.1109/IEEM55944.2022.9989810>.
- [9] S. W. Kim and H. Y. Lee, "A Study on the Accumulation and Use of Corporate Records: Corporate Records Management as a Big Data Platform", Journal of Korean Society of Archives and Records Management, Vol. 20, No. 3, pp. 99-118, Aug. 2020. <http://dx.doi.org/10.14404/JKSARM.2020.20.3.099>.

[10] M. H. Jeong and J. I. Park, "Research on Methodology of Efficient Data Gathering for Manufacturing Information by Sector of Small-sized Enterprises", Proceedings of the Spring Joint Conference of the Korean Institute of Industrial Engineers, Gwangju, Korea, pp. 2449-2452, Apr. 2015.

[11] J. Sadowski, "When data is capital: Datafication, accumulation, and extraction", Big data & society, Vol. 6, No. 1, Jan. 2019. <https://doi.org/10.1177/2053951718820549>.

[12] S. S. Kang, H. S. Kim, C. Y. Lee, and S. R. Kim, "Implementation of Data Management Platform for Aluminum Billet Production Process and Cause Analysis of Defect", Journal of KIIT, Vol. 21, No. 3, pp. 143-151, Mar. 2023. <https://doi.org/10.14801/jkiit.2023.21.3.143>.

[13] Ethan Marcotte, "Responsive Web Design", Ed.: 2e édition. Paris : Eyrolles, 2010.

[14] S. H. Lee and Y. J. Lee, "Implementation of Responsive Web Application for Location-based Semantic Search", Journal of KIIT, Vol. 17, No. 5, pp. 1-12, May 2019. <https://doi.org/10.14801/jkiit.2019.17.5.1>.

[15] P. Brusilovsky, A. Kobsa, and W. Nejdl, "The Adaptive Web: Methods and Strategies of Web Personalization", Springer, 2007.

[16] M. Klymash, I. Tchaikovskiy, O. Hordiichuk-Bublivska and Y. Pyrih, "Research of Microservices Features in Information Systems Using Spring Boot", 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, pp. 507-510, Jul. 2020. <https://doi.org/10.1109/PICST51311.2020.9467911>.

[17] L. Bass, P. Clements, and R. Kazman, "Software Architecture in Practice", Addison-Wesley, Jan. 2012.

[18] S. Ambler, "The Object Primer: Agile Model-Driven Development with UML 2.0", Cambridge University Press, Mar. 2004.

[19] D. Rosenberg and M. Stephens, "Use Case

Driven object modeling with UML: Theory and Practice", Apress, Jun. 2007.

[20] B. O. Mun, D. Y. Kim, D. S. Park, N. Y. Oh, E. Lee, and C. J. Yoo, "UML Self-Driven Learning Web Application Based on Step-by-Step Instruction and Exercise", The Proceedings of the 2019 KIIT·DCS Summer Conference, Gwangju, Korea, pp. 210-213, Jun. 2019.

[21] H. E. Eriksson and M. Penker, "Business modeling with UML", New York 12, 2000.

[22] M. Jones, J. Bradley, and N. Sakimura, "Json web token (JWT)", No. rfc7519, May 2015.

[23] J. H. Kim and S. B. Park, "Enhancing Anonymization Techniques in Dynamic Database Environments through Allowing Public Tuples for Data Availability", Journal of KIIT, Vol. 8, No. 4, pp. 101-113, Apr. 2010.

저자소개

김 효 준 (Hyo-Jun Kim)



2021년 2월 : 군산대학교  
컴퓨터공학과(공학사)  
2022년 3월 ~ 현재 : 전북대학교  
대학원 융합기술경영학과  
석사과정  
관심분야 : 웹 기반 시스템,  
소프트웨어 도메인 모델링

유 철 중 (Cheol-Jung Yoo)



1982년 2월 : 전북대학교  
전산통계학과(이학사)  
1985년 8월 : 전남대학교  
계산통계학과(이학석사)  
1994년 8월 : 전북대학교  
전산통계학과(이학박사)  
2012년 1월 ~ 2013년 7월 :

University of California, Irvine(UCI) 국외연구교수  
1997년 1월 ~ 현재 : 전북대학교 소프트웨어공학과 교수  
관심분야 : 소프트웨어 개발 프로세스, 소프트웨어 품질,  
소프트웨어 테스트, 컴포넌트 기반 소프트웨어,  
소프트웨어 도메인 모델링, 빅데이터 분석, 스마트팜