

<http://dx.doi.org/10.17703/JCCT.2023.9.6.1173>

JCCT 2023-11-140

프로젝트 작업 단위 기반 일정 관리 애플리케이션의 설계 및 구현

Design and Implementation of a Project Work Unit-based Scheduling Application

김보민*, 김민영**

Bomin Kim*, Minyoung Kim**

요약 현대 사회에서는 효율성을 강조하여 조직 내 팀 프로젝트와 개인 업무를 세부 계획으로 이끄는 경향이 있다. 24시간 일상에서 시간을 효과적으로 활용하는 능력은 성실함의 지표로 여겨지며, 사람들은 계획적이고 조직적인 사람들을 모방하고 다양한 시스템을 활용하여 일상을 관리하려고 노력한다. 주어진 업무를 효율적으로 수행하려는 이유는 정해진 기간 내에 작업을 완료함으로써 프로젝트의 성패에 영향을 미치기 때문이다. 프로젝트 일정을 소홀히 할 경우, 성공적인 결과를 위협하는 큰 위험으로 여겨집니다. 이는 조직의 대형 프로젝트뿐 아니라 개인의 생활에서도 같이 적용되며, 시간 기반의 효율적인 관리를 강조하는 다양한 일정 관리 도구를 활용한다. 프로젝트 관리에서는 단순히 '오늘'을 기준으로 하는 것이 아니라 세부 업무 진행 상황을 주의 깊게 파악하는 것이 필요하다.

본 논문에서는 우리는 사용자가 속한 프로젝트에 날짜가 아닌 업무 단위로 접근하여 일정을 관리할 수 있는 안드로이드 애플리케이션을 제안하고 그 구현 내용을 소개한다. 본 논문에서 우리가 구현한 애플리케이션은 사용자가 속한 프로젝트를 등록하여 프로젝트의 목표 및 일정을 관리할 수 있다. 더불어, 프로젝트 전체 또는 개별 목표 진행률을 가시화하여 업무에 효율적으로 접근할 수 있는 기능을 제공한다. 이로써 사용자들은 본 논문에서 우리가 제안하는 애플리케이션을 활용하여 본인이 속한 프로젝트에 전념하며 업무 단위로 일정을 관리함으로써 프로젝트의 전반적인 효율을 향상할 수 있다.

주요어 : 프로젝트, 일정 관리, 진행률, 가시화, 애플리케이션

Abstract In modern society, there is a tendency to emphasize efficiency and lead to detailed planning of team projects and individual tasks within organizations. In a 24-hour routine, the ability to use time effectively is considered an indicator of conscientiousness, and people try to imitate planned and organized people and utilize various systems to manage their daily lives. The reason why you want to perform a given task efficiently is because it affects the success or failure of the project by completing the task within a set period of time. Neglecting the project schedule is considered a major risk that threatens a successful outcome. This applies not only to large-scale organizational projects but also to individual life, and utilizes a variety of schedule management tools that emphasize time-based efficient management. In project management, it is necessary to carefully understand the detailed work progress rather than simply based on Today.

In this paper, we propose an Android application that can manage schedules by accessing the user's project in units of tasks rather than dates, and introduce its implementation. The application we implemented in this paper can manage the project's goals and schedule by registering the project to which the user belongs. In addition, it provides the ability to approach work efficiently by visualizing the progress of the entire project or individual goals. As a result, users can use the application we propose in this paper to focus on their projects and manage schedules by task, thereby improving the overall efficiency of the project.

Key words : Project Work, Schedule Management, Progress Rate, Visualization, Application

*준회원, 동의대학교 ICT융복합연구소 학생연구자 (제1저자)

**정회원, 동의대학교 ICT융복합연구소 조교수 (교신저자)

접수일: 2023년 10월 7일, 수정완료일: 2023년 10월 19일

게재확정일: 2023년 11월 10일

Received: October 7, 2023 / Revised: October 19, 2023

Accepted: November 10, 2023

**Corresponding Author: kmyco@deu.ac.kr

Research Institute of ICT Fusion and Convergence, Dong-eui Univ., Republic of Korea

I. 서 론

효율을 중요시하는 현대 사회에서 사람들은 회사나 학교 같은 조직에서 이루어지는 각종 팀 프로젝트는 물론이고 개인이 혼자서 진행하는 업무까지도 세부 계획을 세운다. 한 사람에게 주어지는 ‘하루’는 24시간으로 한정되어 있으며, 이를 효율적으로 다룰 힘이 성실함의 척도로 사용되곤 한다. 따라서 사람들은 계획적인 사람들을 본받아야 할 대상으로 삼고, 시중에 존재하는 다양한 시스템을 활용하여 본인의 일상을 계획적으로 관리하기 위해 노력한다.

이토록 사람들이 계획적으로 업무를 수행하기 위해 힘쓰는 이유는 정해진 시간에 맞추어 업무를 진행하고 완료하는 것이 프로젝트를 성공시킬 수 있는 큰 요인이 되기 때문이다. 프로젝트 일정 관리를 등한시하는 경우, 프로젝트의 성패를 좌우하는 커다란 위협 요소로부터 위협을 받을 수밖에 없다. 이는 회사나 조직에서 진행되는 대형 프로젝트뿐 아니라 개인의 일상에서도 마찬가지로 통용된다. 따라서 일정을 관리할 수 있는 다양한 프로그램이나 수기로 작성할 수 있는 다이어트들을 살펴보면 관리의 기준은 대체로 ‘시간’에 맞춰져 있는 것을 볼 수 있다. 하루인 24시간을 얼마나 효율적으로 활용했는지, 일주일 혹은 한 달을 얼마나 효율적으로 사용했는지를 관리하는 데에 중점을 둔 것이다[1-6].

그러나 프로젝트를 참여하여 업무를 수행하다 보면 관리 기준을 ‘오늘’로 한정 짓는 경우, 프로젝트의 전반적인 측면에서 접근하는 것이 다소 어렵다는 것을 느낄 수 있다. 시간별로 얼마나 효율을 냈는지도 중요하지만, 프로젝트의 세분된 업무들이 어디까지 진행되었는지를 관리하는 것 역시 프로젝트 진행에 있어 중요한 부분이기 때문이다[7-9].

이에 본 논문에서는 일정 관리의 기준을 ‘시간’보다도 ‘프로젝트’에 두어 사용자가 속한 프로젝트가 얼마만큼 진행되었으며 어떠한 상황에 놓여있는지를 효율적으로 관리할 수 있는 시스템을 제안하고, 이를 바탕으로 구현된 Android 애플리케이션을 설명한다. 사용자는 본 애플리케이션을 통해 본인의 일정을 프로젝트 업무 기준으로 관리하고, 그에 따라 프로젝트의 전반적인 사항부터 세부 사항에까지 원활하게 접근하여 업무를 수행하는 데에 집중할 수 있다.

II. 애플리케이션 설계

1. 요구사항 정의

본 논문에서는 프로젝트 일정 관리를 업무 단위로 세분화하기 위해 아래와 같은 요구사항을 정의하고 있다[10-11].

- 사용자는 새로운 프로젝트를 등록할 수 있어야 한다.
- 사용자는 등록된 프로젝트를 목록으로 확인할 수 있어야 한다.
- 사용자는 프로젝트의 상세 정보를 확인할 수 있어야 한다.
- 상세 정보에는 프로젝트 이름 및 기간, 목표, 일정이 포함되어야 한다.
- 사용자는 프로젝트 정보 및 상세 정보를 수정, 삭제할 수 있어야 한다.
- 사용자는 프로젝트 전체 진행률과 목표별 진행률을 확인할 수 있어야 한다.

본 애플리케이션은 일정을 관리할 때 날짜에 기준을 두는 기존의 일정 관리 시스템과 달리 프로젝트 참여자를 대상으로 한정하여 프로젝트를 업무 단위로 나누어 관리한다. 프로젝트와 개별 목표 및 일정들을 블록화하여 목록으로 출력하고, 각 프로젝트나 목표의 진행률을 ProgressBar로 나타내어 프로젝트에 대한 정보를 가시화하는 데에 중점을 두었다[10].

2. 애플리케이션 제공 기능

정의된 요구사항을 토대로 본 애플리케이션에서 제공하는 기능은 (표 1)과 같다. 앞서 언급된 요구사항을 토대로 기능을 세부적으로 나누었다. 본 연구에서 프로젝트를 관리하기 위해 프로젝트 신규등록부터 수정과 삭제 그리고 한 번에 조회할 수 있는 기능을 제공하도록 (표 1)과 같이 분류하였다. 또한, 각 프로젝트에 대한 상세한 정보를 확인하고 그에 맞는 진행률까지 확인하는 기능도 (표 1)과 같이 포함하였다.

표 1. 본 서비스에서 제공될 주요 서비스 내용
Table 1. Main service contents to be provided in this service

기능	내용
프로젝트 조회	사용자가 등록된 프로젝트를 목록으로 보여준다.
프로젝트 등록	사용자는 새로운 프로젝트를 등록할 수 있다.
프로젝트 수정	사용자는 프로젝트 정보 및 상세 정보를 수정할 수 있다.
프로젝트 삭제	사용자는 프로젝트 정보 및 상세 정보를 삭제할 수 있다.
프로젝트 상세 정보	사용자는 프로젝트 상세 정보를 확인할 수 있다. 상세 정보는 프로젝트 이름 및 기간, 목표, 일정이 있다.
진행률 조회	사용자는 프로젝트 전체 진행률과 목표 진행률을 확인할 수 있다.

3. 데이터베이스 구축

본 애플리케이션은 Kotlin의 Room 데이터베이스를 이용하여 모바일 기기의 내장 데이터베이스에 정보를 저장한다. (그림 1)은 요구사항을 바탕으로 나타낸 E-R 다이어그램이다. 앞서 언급된 요구사항과 기능제공 항목의 내용에서 저장될 데이터를 추출해 이를 바탕으로 Entity와 Attribute를 (그림 1)과 같이 설정하였고, 이후 Entity의 Attribute에서 상호 관계를 맺어야 할 부분엔 Relationship을 (그림 1)과 같이 설정하였다[11].

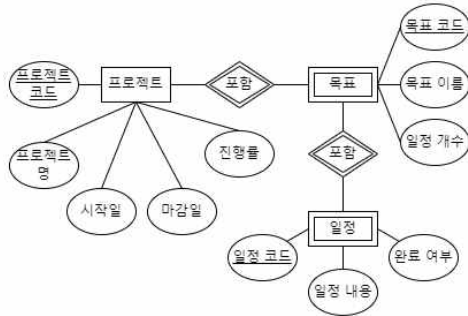


그림 1. E-R 다이어그램
 Figure 1. E-R diagram

(표 2~4)는 E-R 다이어그램을 기반으로 실제 데이터베이스에 구축될 테이블의 명세표에 해당한다 [10-11].

(표 2)는 본 애플리케이션에서 프로젝트에 대한 기본 정보를 담을 수 있는 ‘Project’ 테이블의 명세표이다. 본 애플리케이션에서는 프로젝트 등록 순서대로 관리번호를 부여하고 프로젝트명과 시작일과 마감일 그리고 진행률을 기록할 수 있는 Field로 두었다.

표 2. Project 테이블 명세표
 Table 2. ‘Project’ Table specification table

테이블명	필드명	데이터 타입	기본 값	필드 설명	비고
Project	project_code	Int		프로젝트 코드	PK, AutoGenerate
	project_title	String		프로젝트 명	
	start_day	String		시작일	
	end_day	String		마감일	
	pi_progress	Int	0	진행률	

(표 3)은 본 애플리케이션에서 각 프로젝트에서 달성해야 목표의 데이터를 포함하는 ‘Target’ 테이블의 명세표이다. ‘Project’에 등록된 프로젝트를 기준으로 등록

해야되어 project_code_for Field는 ‘Project’ 테이블의 project_code와 외래키(FK)로 연계된다. 이외의 Filed는 프로젝트 목표에 필요한 목표 명과 이와 연계된 작업 수량을 기록할 수 있는 Filed로 두었다.

표 3. Target 테이블 명세표
 Table 3. ‘Target’ Table specification table

테이블명	필드명	데이터 타입	기본 값	필드 설명	비고
Target	target_code	Int		목표 코드	PK, AutoGenerate
	target_title	String		목표 이름	
	todo_count	Int	0	일정 개수	
	project_code_for	Int		프로젝트 코드	FK

(표 4)는 본 애플리케이션에서 각 프로젝트의 목표별 해야되는 세부 활동의 데이터를 포함하는 ‘Todo’ 테이블의 명세표이다. ‘Target’에 등록된 프로젝트를 기준으로 등록해야되어 target_code_for는 ‘Target’ 테이블의 target_code와 외래키(FK)로 연계된다. 이외의 Filed는 프로젝트 목표에서 실제 수행해야 되는 작업에 대한 이름과 이와 연계된 작업 일정과 해당 작업 달성 여부를 기록할 수 있는 Filed로 두었다.

표 4. Todo 테이블 명세표
 Table 4. ‘Todo’ Table specification table

테이블명	필드명	데이터 타입	기본 값	필드 설명	비고
Todo	todo_code	Int		일정 코드	PK, AutoGenerate
	todo_detail	String		일정 내용	
	end_check	Int	0	완료 여부	
	target_code_for	Int		목표 코드	FK

4. 화면 별 제공 기능

본 애플리케이션은 사용자가 등록한 프로젝트를 목록화하여 조회하고, 새로운 프로젝트 일정을 등록하여 관리할 수 있다. (그림 2)는 애플리케이션의 기능 구조를 도식화했다. (그림 2)에서 도식화된 기준은 사용자에게 제공할 기능을 기준으로 나뉘었다. 각 기능은 안드로이드 애플리케이션에서 하나의 Activity로 제공한다. 각 Activity의 흐름을 (그림 2)에서는 화살선으로 표시해두었다. 그리고 각 Activity에서 제공할 상세 기능은 (표 5)에 기능별 내용을 정리했다[10].

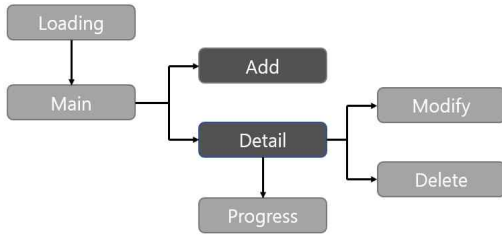


그림 2. 본 애플리케이션의 기능 구조도
Figure 2. Functional structure of this application

표 5. 본 애플리케이션의 Activity 별 제공 기능
Table 5. Functions provided by Activity of this application

Activity명	제공 기능	기타
Loading	로딩 화면 출력	
Main	DB에 접근하여 등록된 프로젝트 블록을 목록으로 출력	
Add	새로운 프로젝트 및 프로젝트에 포함되는 목표, 일정 정보 등록	
Detail	선택된 프로젝트 상세 정보 출력 수정, 삭제 메뉴 (Fragment) 활성화 일정 완료 여부 체크 하여 관리	Main에서 선택된 프로젝트
Progress	프로젝트의 목표별 진행률 출력	Popup Dialog 형 태의 Activity
Modify	프로젝트 정보 수정	Bottom Sheet
Delete	프로젝트 삭제	

III. 애플리케이션 구현

본 애플리케이션은 Android 12버전을 기준으로 구현되었으며, Kotlin을 사용하였다. 앱 실행 확인 및 검증은 LG LM-G900N(VELVET) 장비에서 진행했다.

(그림 3)의 (가)는 로딩 화면으로, 애플리케이션 실행 시 자동으로 가장 먼저 호출되는 화면이다. 3초의 딜레이 시간을 가진 후, 메인화면으로 넘어간다. (그림 3)의 (나)는 로딩 화면에서 넘어온 메인화면이며 등록된 프로젝트를 블록화하여 목록으로 출력한다. 메인화면에서 확인할 수 있는 프로젝트 정보는 프로젝트명, 프로젝트 기간, 진행률이 있다. (그림 3)의 (다)는 메인화면에서 프로젝트 추가 버튼을 클릭하면 실행되는 등록 화면이다. 사용자의 입력에 따라 새로운 프로젝트가 등록된다.

이때, 프로젝트의 이름과 프로젝트 내부 목표의 이름은 공백일 수 없고, 프로젝트 기간의 마감일은 시작일보다 이를 수 없다. 프로젝트 등록이 완료된 경우에 한해 Toast 메시지가 출력되며, 새로운 프로젝트 블록이 갱신된 메인화면으로 돌아간다. (그림 3)의 (라)는 메인화면에서 특정 프로젝트 블록을 선택할 경우, 해당 프로젝트의 상세 정보를 보여주는 상세정보화면이다. 메인화면에서 확인할 수 있었던 정보뿐 아니라 프로젝트에 포함되는 목표와 목표별 일정에 대한 정보를 출력하고, 일정은 CheckBox를 통해 완료 여부에 대한 관리를 수행할 수 있다. 완료된 일정에 따라 프로젝트 진행률을 출력할 때, 해당 일정이 소속된 목표의 전체 일정 개수에 따라 하나의 일정이 가지는 가중치가 다르다. 도움말 버튼을 누르면 이와 관련된 도움말 팝업을 출력한다. (그림 3)의 (마)는 상세정보화면에서 진행률 옆의 더보기 버튼을 누를 시 실행되는 팝업 형태의 Activity이다. 해당 프로젝트의 목표별 진행률을 출력하며, 상세정보화면에서 완료 여부가 갱신되었다면 갱신된 값을 적용해 진행률을 출력한다.

진행률 알고리즘의 의사코드(pseudo-code)는 아래와 같다.

```

val endTodoCount = projectDao?
    .getEndTodoCount(targetCode) //목표의 완료된 일정 수
progressList[i] = (endTodoCount!!.toDouble()
    /todoCount.toDouble()*100).toInt() //목표별
    진행률
val halfIndex = progressList.size/2
val progress =
    progressList.subList(0, halfIndex+1)
    .sum()/itemTarget.size // 프로젝트 진행률
    
```

(그림 3)의 (바)는 상세정보화면에서 메뉴 버튼을 누르면 활성화되는 BottomSheet에서 수정 버튼을 누를 시 출력되는 수정 화면이다. 선택된 프로젝트의 정보를 수정할 수 있도록 EditText 등이 배치되어 있으며, 등록 화면과 마찬가지로 프로젝트명과 목표 이름은 공백으로 둘 수 없다. 수정 버튼을 누르면 DB에 접근해 해당 프로젝트의 정보를 수정하고, 수정값이 갱신된 상세정보화면으로 돌아온다.

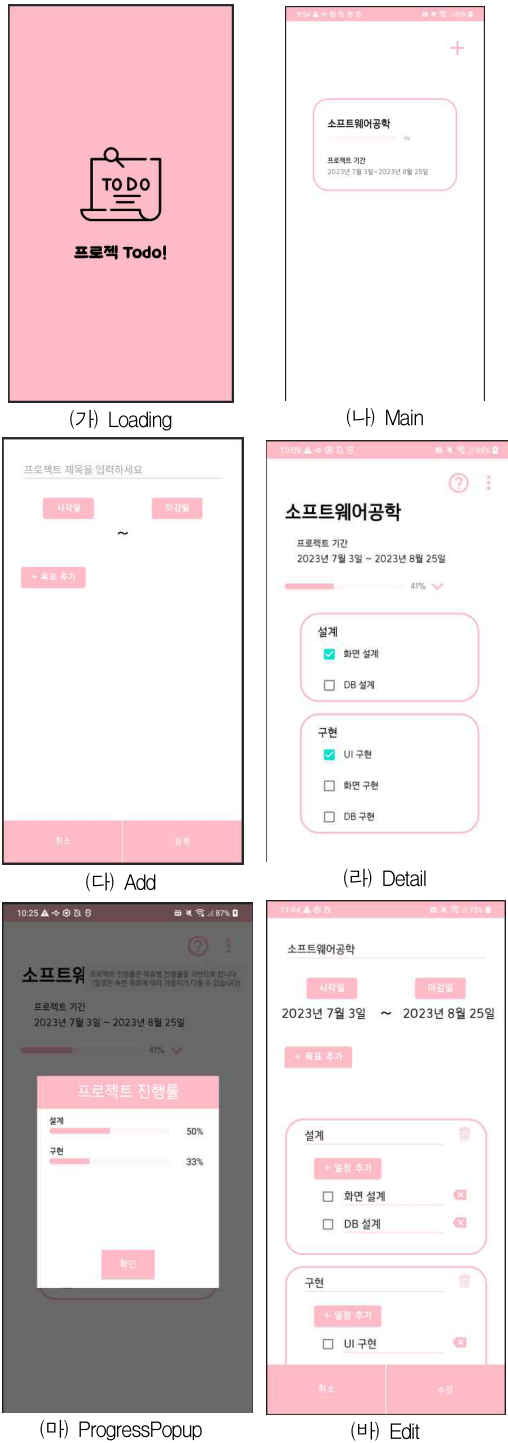


그림 3. 구현된 앱의 각 Activity 실행화면
 Figure 3. Each activity execution screen of the implemented application

IV. 결론

본 연구에서 우리는 프로젝트를 업무 단위로 나누어 관리할 수 있는 시스템을 설계하고 이를 바탕으로 애플리케이션을 구현했다. 우리는 본 연구를 통해 관리의 중점으로 둔 것은 프로젝트 자체이며, 프로젝트의 목표와 일정 등의 세부적인 정보들을 사용자가 다루기 쉽고 한눈에 들어오도록 가시화하는 것이 프로젝트 진행 효율과도 직결되는 것으로 판단했다.

사용자들은 본 애플리케이션을 활용하여 본인이 속한 프로젝트를 업무 단위로 세분화하여 효율적으로 관리하고, 프로젝트 진행 상황을 한눈에 파악할 수 있다. 따라서, 우리는 본 연구를 통해 사용자들이 속한 프로젝트를 보다 효율적으로 관리하여 프로젝트의 전반적 진행 효율을 향상시키는 결과를 기대한다.

References

- [1] G. Shin, W. Yoon, H. Park, W. Jeong, and S. Kim, "Analysis of performance differences according to corporate project management capabilities," Korea Project Management Association 2012 Spring Conference, pp.1-19, Jun 2012.
- [2] S. Kim and W. Yoon, "Analyzing the relationships between project management capability and project performance," Logos Management Review, Vol.13, No.2, pp.139-162, 2015.
- [3] J. Kang, J.W. Kim, J.H. Kim, S. Min, J. Park, G. Lee, J. Jo and K. Kim, "Study on schedule management tools and techniques for project management," Korean Project Management Association 2011 Spring Conference, pp.89-111, Jun 2011.
- [4] Jaeseong Cho, "A Study on Project Schedule Management System Modeling for Small Scale IT Companies," Master's thesis, Department of Industrial Information Management Engineering, University of Ulsan, Korea, 2008.
- [5] Digital News Department of The Digital Times, "Launches 'Dreamy', a smartphone management app that even manages children's schedules" Internet: http://www.dt.co.kr/contents.html?article_no=2015031902109919807058

- [6] E. Park, S. Han, J. Yoon and H. Lim, "Design and Implementation of Scheduler Applications for Efficient Daily Management," *Journal of Internet Computing and Services*, Vol. 22, Issue 2, pp. 41-50, Apr. 2021. DOI : <https://doi.org/10.7472/jksii.2021.22.2.41>
- [7] S. M. Qureshi and C. Kang, "Project Management : Simplifying or Exacerbating," *Society of Korea Industrial and System Engineering*, Vol. 37, No. 2, pp. 9-18, June 2014. DOI: <http://dx.doi.org/10.11627/jkise.2014.37.2.09>
- [8] H. Jeon, K. Park and J. Jung, "A Study on Software Dev. Project Management System Acquisition and Implementing on project site," *Journal of The Korea Society of Computer and Information*, Vol. 20, No. 5, pp. 91-98, May 2015. DOI : <http://dx.doi.org/10.9708/jksoci.2015.20.5.091>
- [9] S. Eom, W. Kim, H. Park and E. Lee, "An Agile applicable for Reading App Development and Design," *Preceding Korean Information Processing Society 2014 Fall Conference Presentation*, pp. 617-620, Nov. 2014.
- [10] M. Kim, J. Ryu, B. Kim, H. Jeong and J. Jang, "Proposal of an Internet Service Providing Information for Consumers to Find the Best Way to Purchase Bottled Water," *The Journal of the Convergence on Culture Technology*, Vol. 9, No. 3, pp. 851-859, March 2023. DOI : <http://dx.doi.org/10.17703/JCCT.2023.9.3.851>
- [11] H. Kim, "IT CookBook, Introduction to Databases (3rd Edition)," *Hanbit Academy*, pp.275-326, 2022.