

# 작업 맥락 및 생산성 추정을 위한 데이터 수집 및 분석 시스템

## Data Collection and Analysis System for Estimating Working Context and Productivity

안단태  
Dantae An

서울대학교  
Seoul National University  
dtan@hcil.snu.ac.kr

서진욱  
Jinwook Seo

서울대학교  
Seoul National University  
jseo@snu.ac.kr

### 요약문

컴퓨터로 작업을 할 때 사용자가 생산적이었는지 파악하기 위해, 사용자의 작업 맥락을 추적하고 작업에 집중하는 시간을 측정하는 방법이 있다. 본 연구에서는 사용자의 작업 맥락으로부터 생산성을 자동으로 추정하기 위해 수집해야 하는 데이터에 대해 탐구한다. 성공적인 작업 맥락 및 생산성 추정에 필요한 데이터의 종류를 정의하고, 해당 데이터를 수집할 수 있는 활동 추적기와 이를 분석하기 위한 시각화 도구를 각각 개발하였다. 그리고 실제 작업 환경에서 데이터를 수집한 후에 세 가지 시나리오를 분석하여 데이터의 적합성을 검증하였다. 한편 같은 작업 맥락에서도 개인마다 느끼는 생산성이 다를 수 있으므로, 특정 작업 맥락이 본인에게 생산적이었는지 기록하는 기능을 시각화 도구에 추가하여 지능형 시스템 학습 시 필요한 생산성 응답 데이터를 수집할 수 있게 하였다. 본 연구는 추후에 사용자의 생산성 향상을 위한 행동 변화를 촉진하는 지능적인 시간 관리 도구를 개발할 때 활용할 수 있다.

### 주제어

작업 맥락, 생산성, 데이터 수집, 시각화, 지능형 시스템

### 1. 서론

컴퓨터로 작업을 하다 보면 누구나 집중이 되지 않는 시간이 있기 마련이다. 그러나 사람들은 본인이 하루에 몇 시간을 작업 외의 활동으로 허비했는지 자각하지 못한다. 이러한 시간을 스스로 인지하고 줄일 수 있다면 작업에 집중하는 시간이 늘어나고 생산성이 향상될 것이다.

자신의 작업 생산성을 돌아보고 시간을 관리하는 방법 중 하나는 일기를 작성하듯이 사후에 자신의 행동과 경험을 떠올리며 돌아보는 것이다. 그러나 지나간 경험을 회상하면서 기록할 때에는 필연적으로 기억의 왜곡과 편향에 의한 현상적 변형(phenomenal transformation)이 발생하게 된다[1]. 따라서 인간의 기억에 의존하지 않고 객관적으로 작업 시간을 측정하도록 돕는 도구들이 등장하였다.

이들은 사용자가 시간 측정 버튼을 눌러 수동으로 데이터를 수집하는 방식과 사용자의 상호작용 없이도 시스템이 자동으로 데이터를 수집하는 방식으로 나눌 수 있다[2]. 타이머 애플리케이션 [3-5]은 대표적인 수동 활동 추적기의 예이다. 이들은 작업을 시작할 때와 마칠 때 타이머를 직접 작동시켜야 하므로 시간 측정에 대한 사용자의 자기 인지(self-awareness)를 필요로 하며[2], 사용성 측면에서 불편한 점이 존재한다.

한편 자동 활동 추적기는 사용자가 작업에 집중하는 동안 의식하지 않아도 사변적(speculative)으로 행동 데이터를 수집한다. 다만 이것이 사용자의 생산성이 향상되도록 행동유도성(affordance)을 부여하기 위해서는 피드백을 적절히 주는 것이 중요하다. 다수의 상용 자동 활동 추적기[6-8]는 데이터 수집 기능과 함께 데이터를 시각화한 대시보드(dashboard)를 제공한다. 하지만 사용자가 사후에 데이터를 관찰하고 분석하면서 본인이 당시에 작업에 집중하고 있었는지 판단하는 활동은 인지적 부담을 수반하기 때문에 대시보드를 통한 피드백 시스템에는 한계가 존재한다. 그리고 이러한 점이 시스템의 접근 부담으로 작용한다[2].

따라서 자동으로 작업 맥락 데이터를 수집하고 이를 바탕으로 사용자의 작업 집중 여부를 추정하여 실시간으로 피드백을 주는 지능형 시스템을 도입할 필요가 있다. 그리고 이러한 시스템을 만들기에 앞서, 본 연구에서는 이것이 학습 기반(learning-based)으로 동작하기 위해 필요한 데이터의 종류를 정의하고 실제로 이를 수집할 수 있는 활동 추적기를 개발하였다. 수집한 데이터를 사람이 보고도 사용자의 당시 작업 맥락을 파악할 수 없다면 지능형 시스템 역시 제대로 된 판단을 내릴 것이라고 기대하기 어렵다. 따라서 본 연구에서는 작업 맥락 시각화 도구 *Working Context Visualization* 을 개발하여 데이터의 적합성을 검증하였다. 한편 사용자의 작업 맥락을 정확하게 파악한다고 하더라도 해당 맥락이 생산적인 시간이었는지 판단하는 기준은 개인마다 다르기 때문에 *Working Context Visualization* 에서는 각 시간대에서의 집중 여부를 사용자 본인이 표기할 수 있는 기능을

제공한다. 이는 지능형 시스템이 작업 맥락으로부터 사용자의 작업 성향(propensity)에 맞게 개인화된(personalized) 생산성을 추정할 수 있도록 하는 기반이 된다.

본 논문의 2 장에서는 작업 맥락을 추정하기 위해 수집하는 데이터의 종류와 이를 수집할 수 있는 자동 활동 추적기에 대하여 다룬다. 3 장에서는 2 장에서 수집한 데이터를 분석할 수 있는 시각화 도구에 대하여 다룬다. 4 장에서는 저자의 작업 환경에서 수집한 데이터를 분석해보며 데이터의 적합성 및 시각화 도구의 사용성을 검증한다. 5 장에서는 본 연구의 결론과 추후 연구 방향을 제시한다.

## 2. 컴퓨터 작업 맥락 추적기

2.1 절에서는 사용자의 컴퓨터 작업 맥락을 추정하기 위해 필요한 데이터의 종류를 정의한다. 그리고 2.2 절에서는 이를 수집할 수 있는 자동 활동 추적기를 소개한다.

### 2.1 수집하는 데이터의 종류

사용자가 컴퓨터로 어떤 작업을 하고 있는지, 그리고 그것이 생산적인 활동인지 파악하기 위해서는 적절한 데이터를 수집해야 한다. 2.2 절에서 제안하는 자동 활동 추적기에서는 실시간으로 다음의 세 가지 데이터를 수집한다.

첫째, 운영체제의 포그라운드(foreground) 프로그램 전환 이벤트를 포착한다. 포그라운드 프로그램이란 동시에 켜져 있는 여러 프로그램 중 전면에 놓여 사용자가 현재 상호작용하고 있는 하나의 프로그램을 말한다. 포그라운드 프로그램이 바뀌는 맥락 전환(context switching)이 일어날 때마다 어떤 프로그램이 새로 전면에 놓였는지 기록한다. 이것은 사용자가 특정 시간대에 수행하고 있던 작업이 무엇인지, 또는 언제부터 언제까지 특정 작업을 수행하였는지 파악할 때 결정적인 정보로 활용된다. 또한 맥락 전환이 일어나는 빈도를 바탕으로 사용자의 작업 집중 여부를 판단할 수도 있다. 수집하는 정보에는 맥락 전환 시각, 프로세스(process) 이름, 그리고 창(window) 이름이 포함된다. 같은 프로그램을 사용할 때 프로세스 이름은 변하지 않지만 창 이름은 사용자의 구체적인 작업 맥락에 따라 달라질 수 있다. 따라서 프로세스 이름은 일반적인 생산성 추정에 쓰이고, 창 이름은 개인화된 생산성 추정에 쓰인다. 한편 웹 브라우저를 사용할 때에는 도메인 이름을 함께 수집하는 것이 도움이 된다[2].

둘째, 키보드의 키 스트로크 이벤트를 포착한다. 이 정보를 통해 사용자가 타자를 치는 작업을 수행하고 있는지 파악할 수 있다. 수집하는

정보에는 입력 발생 시각과 키 종류가 포함된다. 키 종류의 경우 보안 문제를 고려하여 구체적인 문자를 알아볼 수 없도록 “알파벳”, “숫자”, “특수기호” 등으로 비식별화하여 기록한다.

셋째, 마우스의 클릭 및 휠 스크롤 이벤트를 포착하여 상호작용이 발생한 시각과 클릭한 버튼의 종류를 수집한다. 글을 읽는 경우와 같이 키보드를 사용하지 않는 상황에서 마우스 상호작용 발생 빈도를 알면 사용자가 작업에 집중하는 상황과 자리를 비운 상황을 구분할 수 있다. 다만 마우스 이동 이벤트는 수집하지 않기로 하였다. 이를 수집하게 되면 쌓이는 데이터의 양이 급격히 늘어나는 문제가 발생하고, 실시간 화면 캡처 정보 없이 단일 정보로는 작업 맥락을 추정하는 데 도움이 되지 않기 때문이다. 그리고 화면 캡처 정보도 수집하지 않기로 하였다. 데이터 용량이 크고 다른 데이터에 비해 기술적으로 학습과 분석이 어려우며 보안 및 사생활 침해 문제가 발생할 수 있기 때문이다.

### 2.2 자동 활동 추적기

우리는 Windows 운영 체제에서 작업 맥락 추정에 필요한 데이터를 실시간으로 수집할 수 있는 자동 활동 추적기<sup>1</sup>를 구현하였다. 본 추적기는 실행 시 작업 표시줄에 표시되지 않고 시스템 트레이(tray)에 아이콘의 형태로 표시된다. 이는 다른 작업을 하는 동안 사용자가 추적기를 의식하여 방해가 되는 일이 없도록 하기 위함이다. 추적기에서 수집한 데이터는 날짜 별로 정리하여 쉼표로 구분된 값(CSV) 파일로 저장한다.

## 3. 작업 맥락 시각화 도구

자동 활동 추적기로 수집한 방대한 양의 데이터로부터 사용자가 통찰을 얻기 위해서는 이 데이터를 효과적으로 탐색할 수 있도록 하는 시각적 분석 도구가 필요하다. 따라서 그림 1 과 같이 작업 맥락 시각화 도구 *Working Context Visualization*<sup>2</sup>을 개발하였다.

---

<sup>1</sup> <https://github.com/salt26/chordingcoding/tree/v.1.8.3>

<sup>2</sup> <https://github.com/salt26/working-context-visualization>

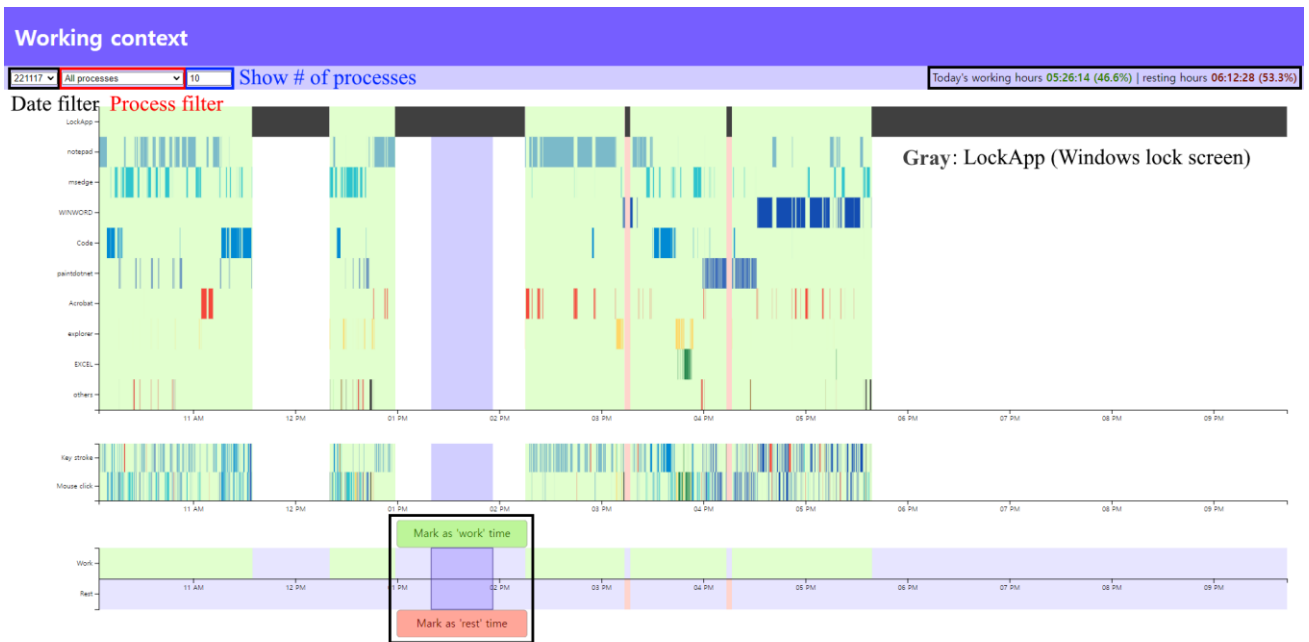


그림 1 Working Context Visualization의 화면. 어느 하루 동안의 전체 시간대를 표시하였다.

화면의 왼쪽 상단에는 시각화할 데이터의 날짜를 변경하는 필터, 특정 프로세스만 보이도록 하는 필터, 그리고 첫 번째 차트에 표시할 프로세스의 개수를 설정하는 필터가 있다. 필터를 통해 불필요한 정보를 숨기고 원하는 정보에 집중하여 살펴볼 수 있다. 오른쪽 상단에는 하루 동안 집중하여 일한 시간, 휴식한 시간 각각의 총합과 이들의 전체 시간 대비 비율을 보여준다. 작업에 집중된 시간보다 작업에 방해되는 활동을 한 시간을 조망하여 보여주는 것이 생산성 향상에 더 도움이 되었다는 선행 연구[2]의 통찰을 반영하여 휴식한 시간을 함께 표시하였다. 이를 보고 사용자는 휴식 시간을 줄이고 작업 집중 시간을 늘리려는 노력을 하게 될 것이다.

그 아래에는 세 개의 차트를 세로로 병치(juxtapose)하였다. 모든 차트는 시간을 나타내는 하나의 가로 축을 공유하고, 한 차트를 확대하는 경우 모든 차트의 시간대(가로 축)가 같은 비율로 함께 확대된다. 따라서 특정 시간대의 작업 맥락 정보를 여러 차트 사이에서 비교하며 보기에 용이하다.

위에서부터 첫 번째에 위치한 차트는 각 시간대에 포그라운드에서 놓였던 프로그램을 열지도(heatmap)로 시각화하며, 세로 축 값이 포그라운드 프로그램의 프로세스 이름이다. 이 시각화를 통해 특정 시간대에 어떤 프로그램들을 번갈아 가며 사용하였는지 직관적으로 파악할 수 있으며, 당시에 어떤 작업을 하고 있었는지 유추하는 것도 가능하다. 총 사용 시간이 긴 프로세스일수록 상단에 위치하도록 정렬하여 사용자가 어떤 프로그램을 가장 오래 사용했는지

확인할 수 있다. 사용 시간의 순위가 표시할 프로세스의 개수 이하인 프로그램들은 모두 마지막 행에 시각화된다. 그리고 각 프로그램을 대표하는 색상을 시각화에 반영하여 범례와 축을 매번 참조하지 않아도 특정 행이 어떤 프로그램을 나타내는지 쉽게 기억하고 떠올릴 수 있도록 하였다. 시각화에 마우스 커서를 올리고 있으면 창 이름 등의 자세한 정보를 툴팁(tooltip)으로 확인할 수 있다.

두 번째 차트는 각 시간대에 일어난 키 스트로크 및 마우스 클릭 이벤트를 보여주는 열지도이다. 이 차트로부터 하나의 포그라운드 프로그램이 장시간 유지된 시간대가 작업에 집중된 시간대인지 자리를 비운 시간대인지 구분할 수 있다. 단일 이벤트를 하나의 사각형에 대응시켜 시각화하면 그 수가 매우 많아 시각화 성능이 떨어지고 각각의 너비가 매우 좁아 잘 보이지 않는 문제가 있다. 따라서 같은 종류의 이벤트가 10 초보다 짧은 간격을 두고 연속으로 발생한 경우 이들을 묶어 하나의 사각형으로 시각화한다. 그리고 단위 시간 당 이벤트 발생 횟수가 많은 묶음일수록 불투명도(opacity)를 높게 설정하여 이벤트 발생 밀도를 시각적으로 알 수 있도록 하였다.

세 번째 차트에서는 사용자가 원하는 시간대를 선택하면 그림 1 하단의 검은색 사각형처럼 해당 시간대에 집중(work)하고 있었는지 휴식(rest)하고 있었는지 표기할 수 있는 인터페이스가 나타난다. 집중 시간으로 표기할 시간대는 모든 차트에서 연두색으로 보이고, 휴식 시간으로 표기한 시간대는 빨간색으로 보이며,

현재 선택한 영역의 시간대는 시스템의 대표색인 연보라색으로 보인다. 집중 여부를 명시적으로 표기하지 않은 모든 시간대는 휴식 시간에 합산된다. 이 기능을 통해 사용자로부터 생산성 응답을 수집할 수 있고, 나중에 데이터를 열람할 때 집중한 시간대와 휴식한 시간대를 한눈에 확인할 수 있다.

#### 4. 평가

여기서는 세 가지 실제 시나리오를 잡고 각 상황에서 주어진 데이터만으로 사용자의 생산성을 판단해보는 사고실험(thought experiment)을 진행한다. 사용자가 특정 시간대에 생산적이었는지 아닌지 추정하는 지능형 시스템 알고리즘의 동작 원리를 사변적으로 따라가면서 앞서 수집한 데이터가 생산성 추정에 적합한지 검증한다. 우리는 7 일 간 실제 작업 환경에서 데이터를 수집하였다. 그리고 *Working Context Visualization* 을 통해 수집한 데이터의 적합성을 잘 보여줄 수 있는 사례를 선별하였다.

그림 1 의 시나리오는 하루 동안의 전체 시간대를 보면서 사용자가 작업에 집중하지 않았던 시간대를 추정하는 상황이다. 첫 번째 차트의 첫 번째 행에 어두운 회색으로 표시된 *LockApp* 이 포그라운드에 있는 동안에는 어떠한 키보드 및 마우스 입력도 없었음을 두 번째 차트에서 확인할 수 있다. 그리고 포그라운드 프로그램의 전환이 장시간 동안 이루어지지 않은 것으로 보아 이 시간대에는 사용자가 자리를 비웠음을 알 수 있다. 생산성 추정 알고리즘이 이렇게 자리를 비운 상황에서의 데이터 패턴을 학습하여 이 시간대가 비생산적인 시간대라고 추정할 수 있다.

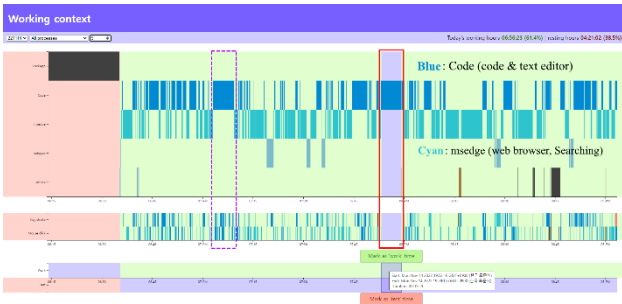


그림 2 코드 편집기(파란색)와 웹 브라우저(청록색)를 번갈아 가며 사용하는 상황.

그림 2 의 시나리오는 특정 시간대에 파란색의 코드 편집기와 청록색의 웹 브라우저를 번갈아가며 사용하는 상황을 보여준다. 왼쪽의 보라색 점선 사각형으로 표시한 시간대에는 코드 편집기에서 키보드 및 마우스 입력이 끊임없이 발생하였기 때문에 작업에 집중하고 있었다고 추정할 수 있다. 그러나 오른쪽의 빨간색 실선 사각형으로 표시한 시간대에는 똑같이 코드

편집기가 포그라운드에 놓여 있지만 아무런 입력이 발생하지 않았다. 이 시간대가 자리를 비운 상황인지, 코드 또는 글을 읽고 있어서 마우스 클릭 이벤트가 없었던 상황인지 주어진 데이터만으로는 구분할 수 없다. 이때 마우스 휠 스크롤 정보가 추가되면 이를 구분할 수 있다. 즉, 활동 추적기가 마우스 휠 스크롤 정보를 추가로 수집해야 한다는 것을 사고실험으로 확인하였다.

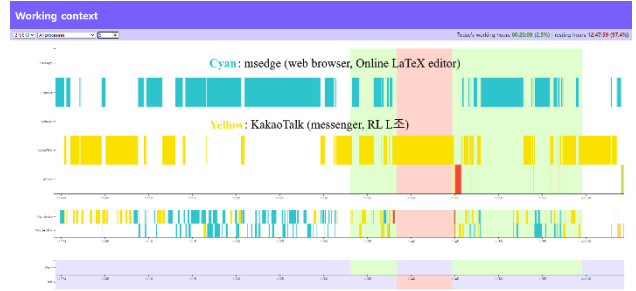


그림 3 웹 브라우저(청록색)와 메신저(노란색)를 번갈아가며 사용하는 상황.

마지막으로 그림 3 의 시나리오는 자정 직전의 1 시간 동안 청록색의 웹 브라우저와 노란색의 메신저 애플리케이션을 번갈아가며 사용하는 상황이 생산적이었는지 판단하는 시나리오이다. 프로세스 이름만 주어진다면 이 시간대가 비생산적인 시간대라고 추정할 가능성이 높지만, 실제로는 사용자가 생산적으로 느낀 시간대였다. 마감 시간을 앞두고 팀 프로젝트를 하면서 웹으로 보고서를 작성하며 메신저로 팀원과 소통하는 상황이었기 때문이다. 이는 각 프로그램의 창 이름을 보면 알 수 있다. 따라서 활동 추적기는 포그라운드 프로그램의 창 이름을 함께 수집해야 한다.

한편 같은 프로세스를 사용하고 있더라도 어떤 상황에서는 생산적이고 어떤 상황에서는 생산적이지 않을 수 있다. 그리고 개인에 따라 특정 작업 맥락이 생산적이었는지에 대한 의견이 다를 수 있다. 그러므로 사용자가 이 상황을 생산적이라고 생각하는지 응답한 데이터가 확보되어야 한다. 본 시각화 도구를 사용하면 사용자가 직접 본인이 생산적이라고 느꼈던 시간대를 표기할 수 있으므로 자동 활동 추적기만으로는 수집할 수 없었던 생산성 응답 데이터를 확보할 수 있다.

#### 5. 결론

우리는 사용자의 컴퓨터 작업 맥락을 자동으로 추적하는 시스템과 여기서 수집한 데이터를 시각화하는 시스템을 구현하였다. 그리고 실제 작업 환경에서 데이터를 수집한 후 분석해봄으로써 수집한 데이터의 종류가 적합한지 평가하였다. 그 결과, 기본적으로는 포그라운드

프로세스 이름과 키 스트로크 및 마우스 클릭 이벤트를 수집하는 것이 도움이 되지만, 글을 읽는 상황과 자리를 비운 상황을 구분하기 위해 마우스 휠 스크롤 정보가 필요하고, 메신저 애플리케이션처럼 용도에 따라 생산성이 달라지는 경우 프로그램의 창 이름과 함께 개인이 당시에 생산적인 활동을 하고 있었는지 응답한 데이터가 필요하다는 사실을 확인하였다.

추후 연구에서는 본 연구의 자동 활동 추적기로 수집한 작업 맥락 및 생산성 데이터를 바탕으로, 사용자의 생산성을 실시간으로 파악하고 생산성이 낮아지는 경우 사용자에게 즉각적으로 피드백을 줄 수 있는 지능형 시간 관리 도구를 구현하고자 한다. 이러한 시스템은 *Working Context Visualization* 을 비롯한 대시보드 형태의 시각화 도구가 사용자의 인지적 부담을 유발한다는 문제점을 효과적으로 해결할 것으로 기대한다. 그리고 지능형 시스템이 지속적으로 학습하기 위해서는 꾸준한 생산성 응답 데이터 수집이 중요하다. 이를 위하여 단순히 응답을 기록할 수 있는 기능을 제공하는 것에 그치지 않고, 선제적(preemptive)으로 사용자에게 현재 생산적인 작업에 집중하고 있는지 묻는 것처럼 행동유도성이 강한 시스템을 고안할 필요가 있다.

## 사사의 글

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF2019R1A2C208906213).

## 참고 문헌

1. Coxon, I. R. Fundamental aspects of human experience: A phenomeno (logical) explanation. In *Experience design: concepts and case studies*. Bloomsbury Academic, 2015.
2. Kim, Y.-H., Jeon, J. H., Cheo, E. K., Lee, B., Kim, K. and Seo, J. TimeAware: Leveraging Framing Effects to Enhance Personal Productivity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 272-283, 2016.
3. Attainr. <https://attainr.com/> Retrieved November 17, 2022.
4. Pomofocus. <https://pomofocus.io/> Retrieved November 17, 2022.
5. Pallo Inc. 열정폼은타이머. <https://pi.tgclab.com/> Retrieved November 18, 2022.
6. Many Tricks. Time Sink. <https://manytricks.com/timesink/> Retrieved June 29, 2021.
7. Apple Inc. Screen Time. <https://support.apple.com/en-us/HT210387> Retrieved June 29, 2021.
8. RescueTime. <https://www.rescuetime.com/> Retrieved November 18, 2022.