

<https://doi.org/10.7236/JIIBC.2024.24.6.21>

JIIBC 2024-6-4

AI를 활용한 간결한 메모와 공유 시스템: QuickQuick

Concise Memo and Sharing System using AI: QuickQuick

윤단비*, 이동건*, 전아린*, 김은비*, 하여린*, 황기태**

Dan-Bi Yoon*, Dong-Geon Lee*, Ah-Rin Jeon*,
Eun-Bi Kim*, Yeo-Rin Ha*, Kitae Hwang**

요약 오늘날 많은 업무에서 컴퓨터와 네트워크를 활용해 다양한 메모와 데이터가 기록되고 공유되고 있다. 그러나 이러한 목적의 기존 도구들은 일시적이거나 간단한 메모를 저장하고 공유하는데 있어 사용자의 시간 소모와 비효율성을 초래한다. 본 논문에서는 이러한 문제를 개선하기 위해 새로운 메모 작성 및 공유 시스템 QuickQuick을 설계하고 구현하였다. QuickQuick 시스템은 회원 가입과 로그인을 자동으로 진행하며, 최소화된 메뉴를 가지고 content-aware 기능을 이용하여 자동으로 메뉴 출력하며, 다른 사용자와 노트 및 데이터를 쉽게 공유할 수 있게 한다. 본 논문에서 구현된 content-aware 기능은 메모 내용을 분석해 자동으로 필요한 메뉴를 즉각 출력하고, 긴 메모를 주제에 따라 자동 분할한다. 본 논문은 제안된 시스템의 활용성을 평가하기 위해 실측하는 방법으로 성능을 평가하였다. 그 결과, 한 사용자가 메모와 25MB의 첨부 파일을 10명과 공유할 때 10명의 사용자에게 전송하는데 약 0.5초 걸리는 것으로 평가되어 본 시스템은 현실에서 실용성이 있는 것으로 판단된다.

Abstract Today, in many tasks, various memos and data are recorded and shared using computers and networks. However, existing tools for this purpose consume users' time and are inefficient in storing and sharing temporary or simple notes. In this paper, we designed and implemented a new memo writing and sharing system, QuickQuick, to improve this problem. The QuickQuick system automatically processes membership registration and login, has a minimized menu, automatically prints the menu using the content-aware function, and allows you to easily share notes and data with other users. The content-aware function implemented in this paper analyzes the contents of the memo, automatically outputs the necessary menu immediately, and automatically divides long memos according to topic. This paper evaluated the performance of the proposed system using actual measurements to evaluate its usability. As a result, when one user shares a memo and a 25MB attached file with 10 people, it is estimated that it takes about 0.5 seconds to transmit them to 10 users, so we conclude that this system is practical in reality.

Key Words : Content-Aware, Memo, AI, Contents Share

*준회원, 한성대학교 컴퓨터공학부

**종신회원, 한성대학교 컴퓨터공학부, 교신 저자

접수일자 2024년 9월 24일, 수정완료 2024년 11월 4일

게재확정일자 2024년 12월 6일

Received: 24 September, 2024 / Revised: 4 November, 2024 /

Accepted: 6 December, 2024

***Corresponding Author: calafk@hansung.ac.kr

Dept. of Computer Engineering, Hansung University

I. 서 론

컴퓨터가 오늘날 사람들의 삶을 편리하게 하였으며 급기야 업무의 대부분은 컴퓨터를 활용하기 때문에, 많은 시간을 컴퓨터 앞에서 보내고 있다[1]. 컴퓨터와 네트워크 기술의 발전으로 많은 사람들과 쉽게 연결되고 정보를 주고받게 되었지만, 이로 인해 우리의 삶은 더욱 바빠지게 되었다.

특히, 간단한 메모를 비롯하여 기억하고 타인과 공유할 일들이 쏟아지고 있다[2,3,4]. 메모에 관한 사례를 보면, 약속이나 갑자기 생각난 처리할 일, 지시받은 간단한 작업, 인터넷 검색을 하다가 발견한 좋은 정보, 스크린 캡처, 기억을 잘 하기 위한 음성 녹음 등 다양하며, 공유에 대한 사례를 보면, 문자 메시지 전송, 팀에 대한 공지, 동료에게 파일 보내기, 음성으로 녹음하여 보내기, 스크린 캡처 보내기, 우연히 발견한 사진 보내기 등 텍스트에서 이미지, 영상, 오디오에 이르기까지 매우 다양하다.

한편, 이들 데이터는 단순하고 크기가 작을 뿐 아니라 일시적인 특징을 가진다. 그러므로 기억할 상황이 해제되면 더 이상 필요 없으며, 어떤 데이터는 잃어버려도 될 만큼 중요하지 않다. 그럼에도 불구하고 컴퓨터 앞에서 업무를 보는 도중에 이러한 데이터가 발생하는 경우, 이들을 파일에 저장하려면, 탐색기와 같은 프로그램을 실행하고, 저장할 위치를 정하고, 파일에 이름을 붙이는 등의 작업은 시간 소모를 많이 가져오며 현재 하고 있는 업무를 방해하는 비효율성을 높이는 문제점이 있다.

현재, 이러한 일시적인 메모를 저장하거나 타인과 공유해야 하는 상황에서, 메모장 앱이나, 스티커 앱, 메모장, 카카오톡의 '나에게 보내기' 등이 활용되고 있다. 하지만 메모장 앱은 텍스트 메시지만 입력하는 한계가 있고, 파일 이름을 붙이고 파일을 적당한 경로에 저장하는 수고가 필요하다. 스티커 앱은 간단한 메모에는 적합하지만, 정보를 타인과 공유하는 기능은 없다. 카카오톡의 경우 기본적으로 타인과의 데이터 공유가 가능하고, '나와의 채팅' 기능을 활용하면, 간단한 메모 작성이 가능하지만, 메모가 체계적으로 분류되지 않고, 대화 창이 분산되어 관리의 어려움이 있다.

본 연구에서는 간단하고 일시적인 데이터를 메모하고 공유하는 방식의 어려움을 해소하기 위해, 보다 효율적인 정보 관리 앱 퀵퀵(이하 QuickQuick)을 설계하고 개발하였다. QuickQuick 시스템의 설계 목적과 해결 기술은 다음과 같다.

첫째, 메모와 공유 기능을 통합한다.

둘째, QuickQuick 앱의 메뉴를 최소화하는 등 UI를 단순화하고, 사용자가 메뉴 창을 열고 메뉴를 선택하는 등 동선을 단순화함으로써 사용자의 시간 소모를 줄인다. 이 목적을 달성하기 위해 본 논문은 content-aware 기능[5]을 도입하여 사용자의 메모 작성과 공유 처리의 의도를 분석하고 메뉴를 자동으로 출력하는 등 자동화를 구현한다.

셋째, 데스크톱이나 스마트폰 버전으로 별도 개발하지만, 한 개인에 대해서는 데스크톱과 스마트폰을 커플링(coupling)시켜 언제 어디서나 공유되게 한다.

II. 시스템 설계 및 구현

1. 시스템 구조

가. 시스템 구성 요소

본 논문에서 제안하고 구현한 QuickQuick 시스템의 구조는 그림 1과 같이 데이터를 저장하는 서버와 QuickQuick이 실행되는 클라이언트 앱으로 구성된다. QuickQuick 앱은 PC에서 실행되는 데스크톱 용과 스마트폰에서 실행되는 모바일 앱으로 각각 설계되고 구현되었다. 특별히 스마트폰 앱은 Flutter SDK[6]를 사용하기 때문에 안드로이드, iOS를 구분하지 않고 실행된다.

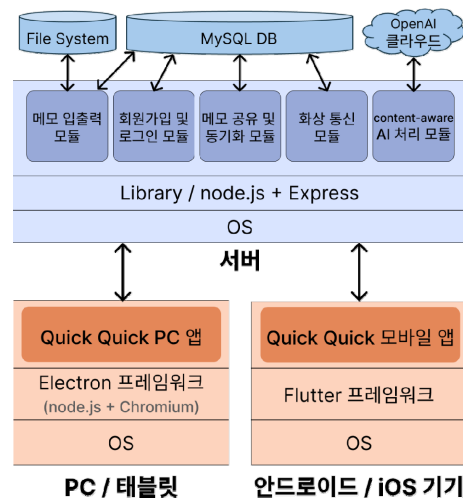


그림 1. 시스템 구조

Fig. 1. Software architecture

데스크톱 용 QuickQuick 앱은 Electron 프레임워크 [7]를 이용하여 자바스크립트와 HTML, CSS로 구현하였다. 그러므로 이 또한 플랫폼에 관계없이 모든 데스크톱

에서 실행된다. 서버는 윈도우 운영체제 위에서 Node.js와 Express 기반의 웹 서비스로 구현되었으며, 메모 데이터의 저장소로 MySQL 데이터베이스를 활용하며, 메모에 첨부된 사진, 동영상 등의 파일은 파일 시스템에 저장한다.

나. 시스템 작동

시스템 내 구현된 소프트웨어는 그림 1과 같이 크게 5개의 모듈로 구성된다. 이들은 회원 가입 및 로그인을 담당하는 모듈, 사용자가 작성한 메모와 사용자가 첨부한 데이터를 저장하고 읽는 메모 입출력 모듈, 여러 사용자 사이에 메모 공유와 한 사용자의 PC와 모바일 기기 사이의 데이터 동기화를 위한 동기화 및 메모 공유 모듈, 화상 통신 모듈, Content-aware 기능을 처리하는 AI 처리 모듈 등이다.

2. 자동 회원 가입 및 자동 로그인

QuickQuick 시스템에서는 사용자의 간편한 사용을 위해 회원 가입과 로그인이 자동으로 이루어지게 하였다. 본 시스템에서는 로그인 이름을 클라이언트에 저장해두지 않고 사용자가 자신의 로그인 이름을 기억하지 않아도 된다. 이를 위해 서버에 데이터베이스에는 사용자당 User ID와 로그인 이름의 두 정보를 저장해둔다.

가. 자동 회원 가입

자동 회원 가입이 이루어지는 과정은 다음과 같다. QuickQuick 앱은 실행될 때마다 UUID 형식의 고유한 User ID를 생성한다. User ID는 맥주소(mac address)와 타임스탬프 등 컴퓨터의 하드웨어 정보를 기반으로 생성되기 때문에 동일 기기에서 늘 같은 User ID가 생성되며 이론적으로 다른 User ID와 같은 확률은 거의 없는 정도로 낮다.

QuickQuick 앱은 생성된 User ID를 서버로 보내 서버 내 데이터베이스에 이미 존재하는지 확인한다. 만약 존재하지 않는다면 신규 사용자로 간주한다.

한편, User ID는 버전에 따라 다소 차이가 있지만, 32개의 16진수와 4, 5개의 '-' 문자로 구성되는 UUID 형태이다. 예를 들면 77638b7c-2af7-010d-98cd-d601c3d4c019 등으로 화면에 출력하여 사용자를 표기하는 것은 매우 불편하다.

그러므로 서버는 사용하고 나타내기 편한 문자열로 로그인 이름(User Name)을 한 개 추가 생성하고 User ID

와 User Name을 쌍으로 데이터베이스의 사용자 정보에 해당하는 테이블에 저장한다. User Name은 형용사와 명사로 구성된 각 배열에서 무작위로 선택된 하나의 형용사와 명사를 조합하여 만든다. 예를 들어, '고요한 구름'과 같은 형태이다. 생성된 User Name이 이미 존재하는 경우, 이름 뒤에 0~1000 사이의 숫자를 추가하고 다시 중복 여부를 확인한 후 고유한 User Name으로 할당한다.

서버가 User ID와 User Name을 생성하고 저장하고 나면, 클라이언트 앱으로 User Name만 보낸다. 그 후 서버는 User Name을 통해 자동으로 로그인 과정을 진행한다. QuickQuick 앱이 실행될 때 생성하여 서버로 전송한 User ID가 서버의 데이터베이스에 이미 있는 경우, 서버는 User ID와 함께 저장된 User Name을 알아내고 서버는 사용자 모르게 자동으로 로그인 과정을 진행한다. 그리고 QuickQuick 앱으로 User Name을 보낸다.

나. 자동 로그인

로그인 과정은 QuickQuick 앱이 실행을 시작하고 회원 가입을 확인하는 과정에서 자동으로 진행되므로, 클라이언트에서는 특별히 할 일이 없다. 로그인 과정의 끝에 QuickQuick 앱은 User Name을 서버로부터 전달받는다.

User Name은 추후 다른 사용자가 친구 추가, 메모 전송, 화상 통신 등에서 더 쉽고 직관적으로 해당 사용자를 식별하고 소통할 수 있도록 제공된다. 또한 각 사용자는 마이페이지에서 자신의 사용자 이름을 확인하고, 사용자가 원하는 이름으로 언제든지 변경 가능하다.

결국 본 논문에서 자동 회원 가입과 자동 로그인 기능을 이용하면 QuickQuick 앱이 실행되는 클라이언트 컴퓨터에 로그인 이름 등 로그인 관련 정보를 저장해두지 않아도 되며, QuickQuick 앱이 실행되어 서버에 접속하면 서버에서 자동으로 로그인 과정이 진행되고, 오히려 서버로부터 데이터베이스에 저장된 로그인 이름인 User Name을 전달받는다.

3. 저장

QuickQuick 시스템의 모든 데이터는 서버의 MySQL 데이터베이스와 파일 시스템에 분산 저장한다. MySQL 데이터베이스의 memo 테이블에 User ID, 메모 ID, 메모 제목, QuickQuick 앱이 실행되는 클라이

언트에서의 메모 창 크기(너비와 높이), 스크린 위치(x좌표 y좌표)를 저장하고, 메모 텍스트와 첨부 파일들의 경로명은 서버의 데이터베이스에 저장되고, 첨부된 파일들은 서버의 파일 시스템에 저장된다. 첨부된 파일은 이미지, 동영상 등 모든 문서들이 지원된다.

그리고 QuickQuick 앱에서 작성되는 모든 메모는 별도의 저장 지시 없이 메모를 닫으면 즉각 서버에 저장된다.

4. 메모 작성

사용자는 QuickQuick 앱에서 직접 텍스트를 입력할 수 있을 뿐 아니라, 이미지, 동영상, PDF 등 다양한 형식의 파일을 첨부할 수 있다. 그림 2는 QuickQuick 앱의 PC 버전과 모바일 버전에서 각각 메모를 작성하는 사례이다.



그림 2. QuickQuick 앱에서의 메모 작성
Fig. 2. Writing memo on QuickQuick app

5. 메모 공유

QuickQuick 시스템은 여러 사용자들 사이에 메모를 공유할 수 있게 한다.

가. 다른 사용자와 공유

QuickQuick 앱에서 사용자가 전송 버튼을 누르면, 서버에서 등록된 친구 목록을 가져와 화면에 체크박스 형태로 표시한다. 이 목록은 서버 데이터베이스의 친구 테이블에서 각 친구의 User ID와 User Name을 불러와 배열로 관리하며, 사용자에게는 친구 이름이 표시된다.

사용자는 체크박스를 통해 메모를 전송할 수신자를 선택할 수 있으며, 선택된 친구의 인덱스를 기반으로 해당 수신자의 User ID를 서버로 전송하도록 한다.

확인 버튼을 누르면, QuickQuick 앱은 메모의 모든 내용을 서버 데이터베이스에 저장한 후, 발신자의 User ID, 선택된 수신자의 User ID, 그리고 메모 ID를 서버로 전송한다. 서버는 이 정보를 이용해 메모 ID에 해당하는 데이터를 찾아, 수신자의 User ID 영역에 메모를 복사한다.

메모에 첨부 파일이 포함된 경우, 파일 시스템에 첨부 파일을 복사하고, 수신자의 메모 정보에는 해당 파일의 경로명을 저장한다. 이 과정이 완료되면, 서버는 수신자의 QuickQuick 앱으로 메모 도착 알림을 전송한다. 알림을 받은 수신자는 즉시 메모를 확인할 수 있으며, 메모 확인 시 서버로부터 복사된 메모를 다운로드하여 사용자에게 보여준다.

나. PC 앱과 모바일 앱의 동기화

QuickQuick 시스템에서는 동일한 User ID를 가진 모바일 앱과 PC 앱 간의 데이터 동기화가 이루어진다. 사용자가 모바일 앱에서 PC 앱의 마이페이지에 표시된 QR 코드를 스캔하면, 서버는 QR 코드에 포함된 User ID가 데이터베이스에 존재하는지 확인한다. User ID가 확인되면 로그인이 승인되며, 모바일 기기에서는 shared_preferences 플러그인을 사용하여 Android에서는 SharedPreferences, iOS에서는 UserDefaults 라는 각 플랫폼의 로컬 저장소에 User ID를 저장한다.

이후, 동기화는 사용자가 새로고침을 수행할 때마다 자동으로 이루어지며, 해당 User ID와 연관된 모든 메모 데이터가 PC와 모바일 앱에서 실시간으로 일치하도록 업데이트된다. 이를 통해 사용자는 기기와 관계없이 동일한 메모를 조회하고 편집할 수 있으며, 추가하거나 수정된 내용은 동기화된 모든 기기에 즉시 반영된다.

6. Content-aware 처리

본 연구에서는 메모 콘텐츠를 분석하여 자동으로 메뉴를 생성해주는 등 content-aware 기능을 설계하고 구현하였다.

가. Content-aware 자동 메뉴 추천

QuickQuick 시스템은 앱 메뉴를 최소화하여 사용자가 메뉴 창을 열고 메뉴를 선택하는 등 동선을 단순화함으로써 사용자의 시간 소모를 줄이는 것이 본 논문의 목표

중의 하나이다. 이를 위해 본 논문에서는 content-aware 기능을 도입하였다.

content-aware란 메모에 입력되는 콘텐츠를 분석하여 자동으로 사용자의 의도를 판단하고 이에 적합한 메뉴를 즉각 제공하여 사용자가 바로 원하는 메뉴를 실행할 수 있게 한다. 이를 통해 사용자가 메뉴를 찾아야 하는 번거로움과 시간 소모를 줄일 수 있다. 현재 content-aware 메뉴 자동 추천은 표 1과 같이 7가지 경우를 지원하며 자동 메뉴 추천 알고리즘은 다음과 같다.

[스텝1] 입력된 메모 텍스트를 프롬프트와 함께 OpenAI API(gpt-4)에[8] 전달하여 해당 텍스트에 가장 적합한 카테고리를 표 1의 카테고리 중에서 선택하도록 요청한다.

[스텝2] OpenAI API(gpt-4)로부터 받은 카테고리 정보를 분석하여 해당 카테고리에 맞는 표 1의 메뉴를 자동으로 출력한다.

[스텝3] 사용자가 작성한 메모 문장 하단에 즉각 메뉴를 출력한다. 사용자가 해당 메뉴를 선택하면 이에 해당하는 작업을 처리한다. 예를 들어 “기말고사는 12월 15일” 문장의 경우, 일정이란 카테고리로 분류되고, QuickQuick 앱은 자동으로 “일정”이라는 메뉴를 출력하고 사용자가 이 메뉴를 클릭하면 캘린더 프로그램을 실행시켜 “기말고사는 12월 15일”을 저장한다. 캘린더 내용은 추가로 삽입하거나 편집, 삭제할 수 있다. 또 다른 사례로 “독서 모임 장소: 혜화역”의 메모에 대해서는 “위치” 메뉴가 출력되고 사용자는 메뉴를 클릭하면 그림 3과 같이 카카오맵으로 바로 확인할 수 있다.

표 1. 7가지의 Content-aware 자동 메뉴 추천 사례
Table 1. Seven content-aware automatic menu recommendation cases

카테고리	메모 사례	추천 메뉴
일정	“기말고사는 12월 15일”	일정
위치	“독서 모임 장소: 혜화역”	위치
번역	“I love you.” (외국어인 경우 번역)	번역
URL	https://www.naeil.com/news/read/511320?ref=naver	요약
전송	“아린이에게 전송해”	전송
통신	“은비야 전화하자”	통신
분할	메모 내 여러 주제의 문단들이 있는 경우 요약 혹은 분할	요약, 분할
기타	내 계좌번호: 490-123456-78-901	없음



그림 3. QuickQuick 앱의 위치 확인 기능
Fig. 3. Confirm location function on QuickQuick app

메모 콘텐츠가 여러 카테고리에 해당할 경우, 표 1과 같이 해당되는 메뉴를 모두 추천한다. 특히, 요약 및 번역 기능은 이미 작성된 메모에서도 해당 문단에 커서를 올리면 다시 활성화되어, 메모를 더욱 유연하고 효과적으로 관리할 수 있도록 돕는다.

나. Content-aware 다중 문단 자동 분할

사용자는 하나의 메모에 여러 주제의 내용을 담기도 한다. QuickQuick 시스템은 이런 메모에 대해 OpenAI API(gpt-3.5-turbo)와 OpenAI API(gpt-4)를 통해 내용의 유사성에 따라 여러 개의 메모로 분할하여 관리하고자 하였다. OpenAI API(gpt-3.5-turbo)를 사용하여 메모를 자동으로 여러 개의 독립적인 소주제로 분리하였으며, OpenAI API(gpt-4)를 통해 첨부파일들을 앞서 분리된 소주제에 맞추어 연결시킨다. 이렇듯 각 모델의 특성에 맞추어 작업을 분리함으로써 작업 효율을 최적화하고 모델 운영 비용을 절감하고자 하였다. 다중 문단을 자동으로 분할하는 알고리즘은 다음과 같다.

[스텝1] 메모 데이터를 OpenAI API(gpt-3.5-turbo)로 보내 단락의 주제가 달라질 때, 단락을 메모에서 분리해 내도록 지시한다.

[스텝2] 분리된 각 단락에서 문장을 분석해 주제를 식별한다. 메모에 URL이 포함된 경우 OpenAI API(gpt-3.5-turbo)를 통해 해당 URL의 HTML 데이터를 가져오고, 페이지의 제목을 추출하여 URL과 메모의 내용 간 연관성을 분석한다.

[스텝3] 메모 분할 후, 첨부된 파일들도 분류된 메모에 연결시키는데, 첨부 파일이 어떤 주제를 가지는지를 분류하는 방식은 표 2와 같다. 첨부 파일의 주제를 판단하기 위해서는 OpenAI API(gpt-4)를 사용한다. 만약 파일이 기존 메모와 관련이 없다고 판단되면, 새로운 메모로 분리한다. 그림 4는 그림 2의 메모를 자동 분류한 결과 3개의 메모로 분할된 것으로 보여준다.

표 2. 다중 문단 분할에서의 파일 형식 및 처리 방법

Table 2. File types and libraries of processing in multiple paragraphs

파일 형식	파일 내 콘텐츠 인식 방식
PDF 파일	PDF 문서에서 텍스트를 추출한다. 만약 파일이 10페이지를 초과할 경우, 첫 10개의 페이지만 fs모듈의 writeFile() 메서드를 이용하여 os모듈을 통해 각 운영체제에서 표준으로 사용하는 임시 파일을 저장하는 디렉토리에 임시로 저장한다. 분할이 끝난 후, fs.unlinkSync() 메서드를 통해 디렉토리에서 해당 파일을 삭제한다.
이미지 파일	이미지에서 텍스트를 추출하여 메모와 연관시킨다.
텍스트 파일	파일에서 텍스트를 직접 읽어온다.
기타 파일	위에서 지원하지 않는 형식의 파일은 파일 이름을 텍스트로 설정해 관련 메모와 연관시킨다.



그림 4. QuickQuick 앱의 분류 기능

Fig. 4. Classification function on QuickQuick app

7. 화상 통신

본 연구에서는 그림 5와 같이 QuickQuick 시스템에 공유의 기능을 확장하여 실시간으로 다른 사용자와 화상 통신할 수 있는 기능을 구현하였다.

화상 통신 기능은 WebRTC API[9]를 활용하여 구현하였으며, 여러 명이 동시에 통신할 수 있게 하였다. 화상 통신 중 Google Speech-to-Text API를 이용하여 자동 음성 변환 기능을 제공하므로 대화가 텍스트로 실시간 변환된다.



(a)PC앱에서의 화상통신

(b)모바일 앱에서의 화상통신

그림 5. 화상 통신 화면

Fig. 5. Video communication screen

III. 성능 평가

1. 성능 평가 목적과 대상

본 논문은 시스템의 실용성을 평가하기 위해 성능을 평가하였다. 본 시스템의 가장 핵심 기능은 메모를 저장하거나 메모를 다른 사용자와 공유하는 것이기 때문에, 본 성능 평가는 사용자가 스스로를 위해 메모를 저장하는 성능보다 메모나 데이터를 여러 사람에게 전송하는 공유 기능에 초점을 맞추고, 한 사용자가 파일을 다른 사용자에게 공유하는데 걸리는 시간을 측정하고 평가하였다.

2. 공유 대상자 수에 따른 공유 속도 평가

본 실험은 파일의 크기 및 공유 대상자의 수에 따른 파일 전송 속도를 평가하는 것을 목적으로 한다. 시간 측정은 파일 크기와 공유 대상자 수를 다음과 같이 바꾸면서 진행되었다.

- 파일 크기: 500KB, 1MB, 5MB, 8MB, 25MB, 100MB
- 공유 대상자 수: 1명, 3명, 5명, 7명, 10명

파일 공유 시간을 측정한 결과 공유 시간은 그림 6과 같다.

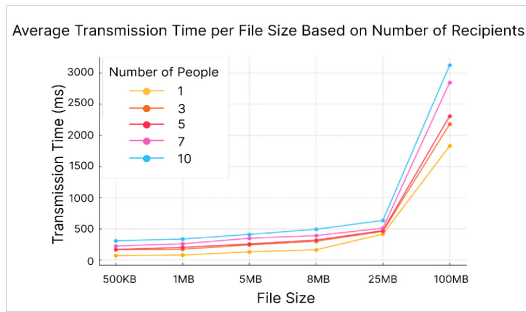


그림 6. 공유 대상자 수에 따른 성능 평가
Fig. 6. Average Transmission Time per File Size Based on Number of Recipients

실험 결과, 실험 시점에 네트워크 상황에 따라 약간씩 전송 시간이 달라질 것으로 판단되지만, 전반적으로 파일의 크기가 커질수록, 그리고 공유 대상자 수가 많아질수록 전송 시간이 선형적으로 증가하는 양상을 보였다. 파일 크기가 25MB 이하인 경우, 공유 대상자가 10명에 이르기까지 거의 0.5초 내에 전송되므로 사용자들은 시간 차이를 거의 느끼지 못하며 불편함 또한 느끼지 못할 것으로 예측된다.

100MB에서 전송 시간이 크게 증가하게 보이는 것은 착시이며, 파일 크기가 25MB에서 급격히 100MB로 증가했기 때문이다. 100MB가 3명에게 전송되는 경우 약 2.2초 정도 걸리는 것으로 평가되어, 100MB 이하의 소규모 파일이나 소수의 대상자에게 전송할 때 매우 짧은 시간 내에 전송이 완료되어 실용성이 뛰어나다는 점을 확인할 수 있다.

100MB이상의 경우처럼 파일이 매우 큰 경우 파일 전송시간을 줄이기 위해 데이터 압축 방법 등을 사용하면 시간을 줄일 수 있을 것으로 판단한다. 이 방법은 추후 연구과제로 남겨둔다.

IV. 결 론

본 논문은 AI 기능을 활용하여 메모를 작성하고 다른 사람과 공유하는 작업을 돕는 시스템을 설계한 내용을 소개하였다. 사용자의 편의를 위해 자동 회원 가입 및 자동 로그인 기능을 구현하고, 사용이 쉽고 빠르도록 하기 위해 메뉴를 최소화하고, 메모 텍스트를 자동으로 분석하여 필요한 메뉴를 자동으로 추천하고, 여러 주제를 담은 긴 메모는 여러 개의 메모로 자동 분할하는 Content-aware 기능을 도입하였다. 이 기능은 사용자의 메모 작성에 파

른 시간 소모를 줄인다. 또한 메모에 다양한 형식의 파일을 첨부할 수 있게 하였고, 다른 사람과 메모를 쉽게 주고받을 수 있는 공유 기능을 구현하였다.

본 논문은 제안된 시스템의 활용성을 평가하기 위해, 실측하는 방법으로 메모와 첨부 파일에 대한 공유 성능을 평가하였다. 성능 평가 결과, 10명이 서로 공유할 때 첨부 파일을 포함하여 25MB의 메모에 대해 0.5초 걸리는 것을 평가되었고, 100MB의 메모를 3명이 공유하는 경우 2.2초로 평가되어 본 논문에서 제안되고 구현된 시스템은 현실에 높은 실용성을 가진다고 판단된다.

References

- [1] Ravasio, P., Guttormsen Schär, S., & Krueger, H. "In Pursuit of Desktop Evolution: User Problems and Practices With Modern Desktop Systems", ACM Transactions on Computer-Human Interaction(TOCHI), Vol. 11, No. 2, pp.156-180, JUN.01, 2004.
<https://dl.acm.org/doi/10.1145/1005361.1005363>
- [2] Discover the Benefits of Using a Sticky Notes App on Your Windows Device,
<https://www.consumersearch.com/technology/discover-benefits-using-sticky-notes-app-windows-device>, Dec. 22, 2020
- [3] Lauren Bertino, "The Benefits of Using Online Sticky Notes",
<https://www.hoylu.com/the-benefits-of-using-online-sticky-notes/>, April 30, 2021
- [4] Anuj Gupta, "What are the benefits of using memo pads?",https://paperlla.com/blogs/news/what-are-the-benefits-of-using-memo-pads?srltid=AfmBOopI1-DFOj3dchJ5_GqXLPwnHjqE9v-Ooi-FITwT86n9tW8hQpKy, July 21, 2023
- [5] Huang-Chia Shih, "A Survey of Content-Aware Video Analysis for Sports". IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 5, pp. 1212-1231, 2018,
doi: <https://10.1109/TCSVT.2017.2655624>.
- [6] Hina Hussain, et. al, "Comparative Study of Android Native and Flutter App Development", The 13th International Conference on Internet, pp.99-102, 2021
- [7] Kitae Hwang, et al., "Implementation of CoMirror System with Video Call and Messaging Function between Smart Mirrors", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 22, No. 6, pp.121-127, 2022.
doi: <https://10.7236/JIIBC.2022.22.6.121>.
- [8] Cristian Mauricio Gallardo Paredes, Cristian R. Machuca, Yadira Semblantes, "ChatGPT API: Brief

overview and integration in Software Development”, DOI: <https://doi.org/10.61961/injei.v1i1.7>

- [9] Branislav Sredojev, Dragan Samardzija and Dragan Posarac, “WebRTC technology overview and signaling solution design and implementation”, The 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, pp. 1006-1009, 2015, doi: <https://10.1109/MIPRO.2015.7160422>.

저 자 소 개

윤 단 비(준회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 모바일 시스템, 웹 공학, IoT, 빅데이터

이 동 건(준회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 웹 서버, IoT

전 아 름(준회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 모바일 시스템, 데이터 사이언스

김 은 비(준회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 모바일 시스템, 웹 공학, 빅데이터

하 여 름(준회원)



- 한성대학교 컴퓨터공학부 재학
- 관심분야 : 모바일 시스템, 웹 공학

황 기 태(중신회원)



- 서울대학교 컴퓨터공학과 학사
- 서울대학교 컴퓨터공학과 석사
- 서울대학교 컴퓨터공학과 박사
- 현재 한성대학교 컴퓨터공학부 교수
- 경력
University of Florida 방문 교수
- 관심분야 : 모바일 시스템, IoT, 인공지능

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음.