“NFC 방명록”

* 1. 

|  |  |
| --- | --- |
| 과목명 | 기초창의공학설계 |
| 담 당 | 이 상 윤 교수 |
| 작성일 | 2023년 12월 14일 |
| 프로젝트명/조 | NFC방명록/9조 |
| 조 장 | 정기주(2021428890) |
| 조 원 | 전호준(2021115459) |
|  | 금명섭 (2020111842) |

목차

**1. 프로젝트 개요**

**2. 결과물 및 상세 개발 내용**

2.1 결과물

2.2 세부 개발 내용 및 결과

2.2.1 DB

2.2.2 아두이노

2.2.3 파이썬

2.24 png 파일

**3. 사용자 메뉴얼**

3.1 com번호입력

3.2 회원가입

3.3 메인화면

3.4. 방명록 보기 화면

**4. 문제점**

4.1 인지된 문제점

4.2 문제점 대처

**5. 프로젝트 진행**

5.1 프로젝트 진행 방식

5.2 역할 분담

**1.** **프로젝트 개요**

방명록 프로그램으로 이 프로젝트는 아두이노, 파이썬 GUI, 그리고 데이터베이스를 통합하여 개발하였습니다. 사용자 정보를 효과적으로 기록하고 관리하는 것을 목표로 했습니다. 이 프로그램은 특히 이벤트나 행사 현장에서 참가자들의 정보를 간편하게 수집하고 저장하기 위해 설계되었습니다.

사용자 측면: 행사나 이벤트에 참가하는 사용자는 휴대폰 NFC를 통해 간편하게 등록되며, LED와 부저를 통해 등록 여부를 확인할 수 있습니다. 관리자는 GUI를 통해 사용자 정보를 손쉽게 확인하고 관리할 수 있습니다.

기능적 측면: 프로그램은 직관적으로 사용하기 쉬운 파이썬 GUI를 통해 사용자들이 개인 정보를 수집하고, 아두이노를 활용하여 입력된 정보를 수집합니다. 이 때 프로그램이 정상적으로 인식되었음을 알리는 부저와 LED 장비를 부탁했습니다. 정보들은 안전하게 DB에 저장되어 나중에 필요한 경우에 참고할 수 있도록 합니다. 또한 수집된 정보를 간단하게 확인할 수 있도록 GUI에 목록을 볼 수 있는 기능과 정렬 기능을 추가하였습니다.

|  |  |
| --- | --- |
| **프로젝트 명칭** | NFC 방명록 프로그램 |
| **개발 기간** | 2023. 11. 11 ~ 2023. 12. 11 |
| **개발 수행자** | 정기주, 금명섭, 전호준 |
| **개발 방식** | RFID센서를 필두한 아두이노 장치를 개발하고, 시각적으로 정보를 확인할 수 있는 파이썬 GUI코드를 개발한 후 DB에 파일 형태로 저장할 수 있도록 스키마를 작성 |
| **상세 기능** | - 휴대폰이 RFID 센서에 인식 시 LEF와 피에조 부저를 활용하여 사용자에게 정상처리 여부를 시각적, 청각적으로 알립니다.  - 아두이노와 Python 간의 시리얼 통신을 통해 RFID 센서에서 받은 데이터를 실시간으로 전달하고 처리합니다.  - 관리자는 GUI 기반의 Python 프로그램을 통해 행사 참가자들의 정보를 효과적으로 관리할 수 있습니다.  - 회원가입 기능을 통해 처음으로 인원을 등록할 때 성명 및 소속을 작성할 수 있습니다.  - RFID 센서가 인식되면 회원가입으로 기록된 소속, 성명, 시간, 날짜 정보를 DB에 저장합니다.  - Python 프로그램에서는 방문한 사용자를 추적하고, 가시성을 높이기 위해 사용자 정보를 정렬하는 기능을 구현합니다. |

**2. 결과물 및 상세 개발 내용**

**2.1 결과물**

"NFC 방명록" 프로젝트의 최종 결과물은 다음과 같다.

<표 2.1 결과물 내역>

|  |  |
| --- | --- |
| 장치 및 문서 | 결과물 형태 |
| 아두이노 장치 |  |
| GUI 프로그램 |  |
| 소스코드 및 png파일 | 기초창의공학설계 Github 9조 내 업로드 |
| 프로젝트 진행 시 작성된 문서 파일 | 기초창의공학설계 Github 9조 내 업로드 |
| 최종 결과 보고서 | 결과 보고서 문서 1건 및 소스코드 압축파일 |

**2.2 세부 개발 내용 및 결과**

**2.2.1 DB**

- 개발 환경: SQLite

-DB 서버를 따로 구축하지 않고 파일형태로 사용자 정보를 저장하기로 함.

- 회원 정보를 저장하기 위한 User, 방문 정보를 저장하기위한 List 테이블로 작성

- User 테이블의 Pin\_Num을 기준으로 각 NFC를 구분하고 이를 통해 Name과 Content의 정보를 가져온다

<표 2.2.3-1. DB User 테이블 데이터 내역>

|  |  |  |  |
| --- | --- | --- | --- |
| 데이터 명칭 | 정의 | 형식 | etc |
| Seq | 데이터 정렬을 위한 순서 번호 | INTEGER | ex) 1, 2, 3, ... |
| Pin\_Num | 각 NFC에 내장되어 있는 고유한 ID번호 | TEXT |  |
| Name | 회원가입 시 입력되는 ID번호에 대응 되는 성명 | TEXT |  |
| Content | 회원가입 시 입력되는 ID번호에 대응 되는 소속 | TEXT |  |

<표 2.2.3-1. DB List 테이블 데이터 내역>

|  |  |  |  |
| --- | --- | --- | --- |
| 데이터 명칭 | 정의 | 형식 | etc |
| Seq | 데이터 정렬을 위한 순서 번호 | INTEGER | ex) 1, 2, 3, ... |
| Date | RFID가 인식될 때의 날짜 | TEXT |  |
| Time | RFID가 인식될 떄의 시간 | TEXT |  |
| Name | 회원가입 시 입력된 성명 | TEXT |  |
| Content | 회원가입 시 입력된 소속 | TEXT |  |

**2.2.2 아두이노**

- 개발환경: ArduinoIDE

**2.2.2.1 아두이노 장치**

**-** 프로젝트에 주가 되는 RFID 센서는 성능이 좋더라도 기능적인 부분에서 크게 차 이가 없기에 기존에 배포받은 Uno와 RC522를 사용함

1. Arduino Uno R3

2. RFID RC522

3. 피에조 부저, LED 2개

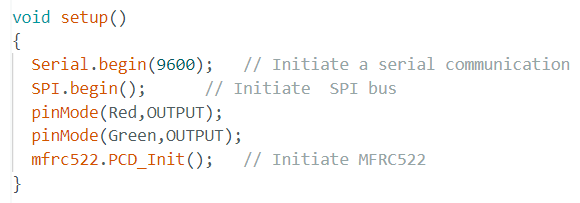
**2.2.2.2 아두이노 코드**

1. 사용 라이브러리: SPI.h, MFRC522.h

SPI 통신 및 RFID 센서를 위한 MFRC522 라이브러리를 사용

2. setup 함수: 시리얼 통신을 시작하고 SPI버스를 초기화한 후, LED핀을 출력으로 설 정(이는 디폴트로 빨간색 LED를 밝히기 위함)하고 MFRC522를 초기화

<그림 2.2.4.2 setup함수>



3. loop 함수: 빨간색 LED를 켜고 초록색 LED를 끄고, 새로운 카드가 감지되면 카드 의 UID를 읽어와서 시리얼 모니터에 출력.(이는 파이썬에서 시리얼 통신에서 가져오 기 위함) 이후 초록색 LED를 켜고 동시에 피에조 부저를 울림

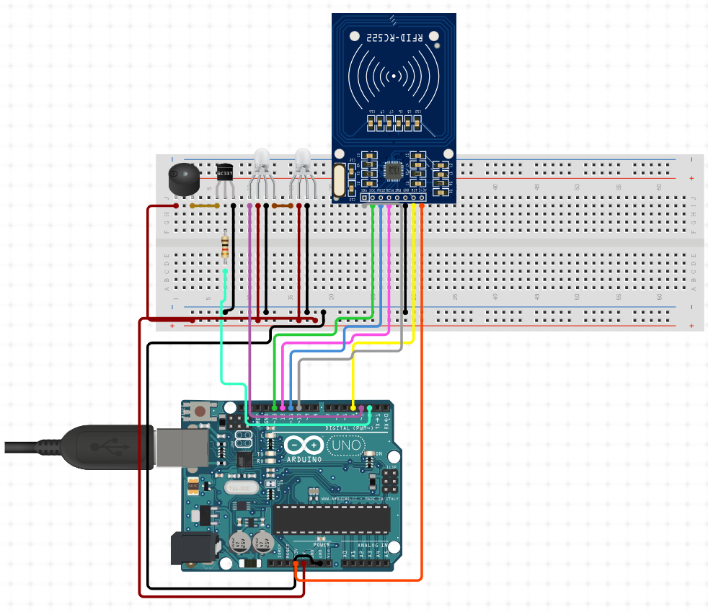
<그림 2.2.4.2 loop함수>



**2.2.2.3 회로도**

- 아두이노의 하드웨어 지식은 전무하기에 https://www.circuito.io/ 사이트를 활용

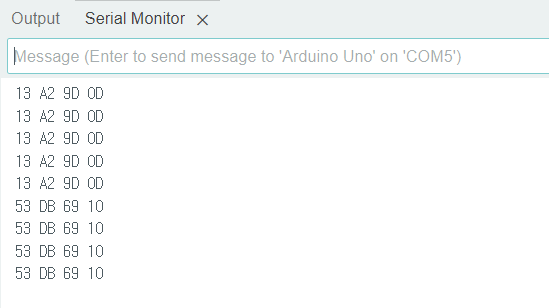
<그림 2.2.4.3 아두이노 회로도>



**2.2.2.4 시리얼 모니터**

- 아두이노 키트의 rfid 태그와 카드를 센서에 인식했을 때 각 고유한 uid가 모니터 에 출력

<그림 2.2.2.4 시리얼 모니터>



**2.2.3 파이썬**

- 개발 환경: 파이참(Pycharm)

**2.2.3.1 db.py**

- SQLite과의 연결을 위해 sqlite3 라이브러리를 활용

<표 2.2.3.1 db.py>

|  |  |
| --- | --- |
| 함수명 | 기능 |
| check\_pin\_num(num) | 핀 넘버를 입력하여 그에 해당하는 모든 정보를 가져와 반환 |
| check\_seq\_user() | User 테이블의 레코드 수를 가져와 반환 |
| get\_last\_user() | User 테이블의 가장 최근 레코드의 Name, Content를 반환 |
| check\_seq\_list() | List 테이블의 레코드 수를 가져와 반환 |
| get\_last\_list() | List 테이블의 가장 최근 레코드의 Date, Time, Name, Content를 반환 |
| get\_list(sort,desc) | sort를 기준으로 오름차, 내림차순으로 정렬한 List 테이블의 모든 레코드를 반환 |
| input\_Name(num, name, content) | Pin번호, name, content를 입력받아 User 테이블에 저장 |
| save(num) | Pin번호를 입력받아 그에 해당하는 Name, content 와 시간 정보를 List 테이블에 저장 |

**2.2.3.2 Make\_label.py**

- 본격적으로 GUI 프로그램을 만들기 전 활용되는 함수를 저장, tkinter를 활용

<표 2.2.3.2 Make\_lable.py>

|  |  |
| --- | --- |
| 함수명 | 기능 |
| image\_label(self, file\_name,x,y) | 파일명, 좌표값을 입력받아 파일명을 기준으로 이미지 객체를 만들고 좌표값이 배치 후 반환 |
| image\_label\_text(self, file\_name, x, y, text, color, font) | 파일명, 좌표, 텍스트, 색, 폰트를 입력받아 이를 적용한 이미지 객체를 만들고 반환 |
| image\_button(self, file\_name, x, y, command) | 파일명, 좌표, 실행시킬 함수를 입력받아 적용한 버튼 객체를 반환 |
| image\_button\_text(self, file\_name, x, y, command, text, color, font) | 파일명, 좌표, 실행시킬 함수, 텍스트, 색, 폰트를 입력받아 적용한 버튼 객체를 반환 |

**2.2.3.3 Guest\_book.py**

- main함수가 담겨있는 파일로 실행시킬 GUI class가 담겨있음

<표 2.2.3.3 Guest\_book.py>

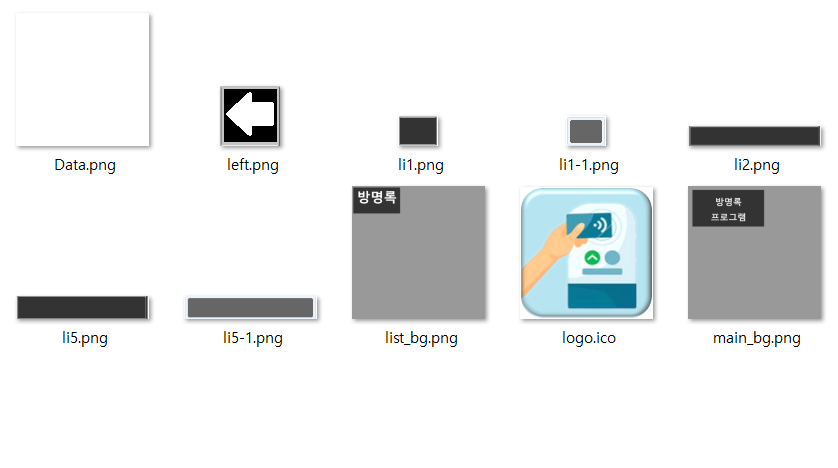
|  |  |
| --- | --- |
| 함수명 | 기능 |
| arduino(self) | thread로 실행되는 함수로 nfc 센서가 인식될 경우에 적절히 처리되도록 작성. com포트 번호를 입력받아 시리얼 통신이 가능케함 |
| destroy(self) | GUI 화면의 위젯들을 삭제함 |
| center\_window(self, width, height) | 높이와 화면을 입력받아 프로그램을 배치 |
| no\_action(self) | 아무 동작도 하지 않은 빈 함수로 특정 상황에서 동작을 하지 않고 넘기기 위함 |
| main\_screen(self) | main 화면을 나타내는 함수로 이미지를 가져와 배치시킨 후 db에서의 최신데이터를 화면에 출력 |
| apply\_screen(self) | 회원가입 화면을 나타내는 함수 |
| commit1,2(self) | 회원가입 화면의 버튼에 적용되는 함수로 '이름'과 '소속'을 입력받고 회원가입이 되도록 함수를 적용 |
| reload(self) | '방명록 보기' 버튼에 적용될 함수로 방명록 리스트로 넘어감 |
| list\_screen1,2,3() | '방명록'의 화면의 수로 최대 3페이지까지 나타냄 |
| sort1,2,3,4(self) | 날짜, 시간, 이름, 소속으로 정렬하게 하는 함수 |
| \_quit(self) | 종료 여부를 확인하는 messagebox를 띄우고 종료를 하는 함수 |
| change\_sortnum(self) | 오름차, 내림차순을 결정하고 다른 함수의 색을 변경하여 시각적으로 확인할 수 있게함 |

**2.2.4 png파일**

- 제작 프로그램: pixlr e

- GUI 프로그램에 사용될 png파일을 제작

<표 2.2.4 제작한 파일-1>



<표 2.2.4 제작한 파일-2>



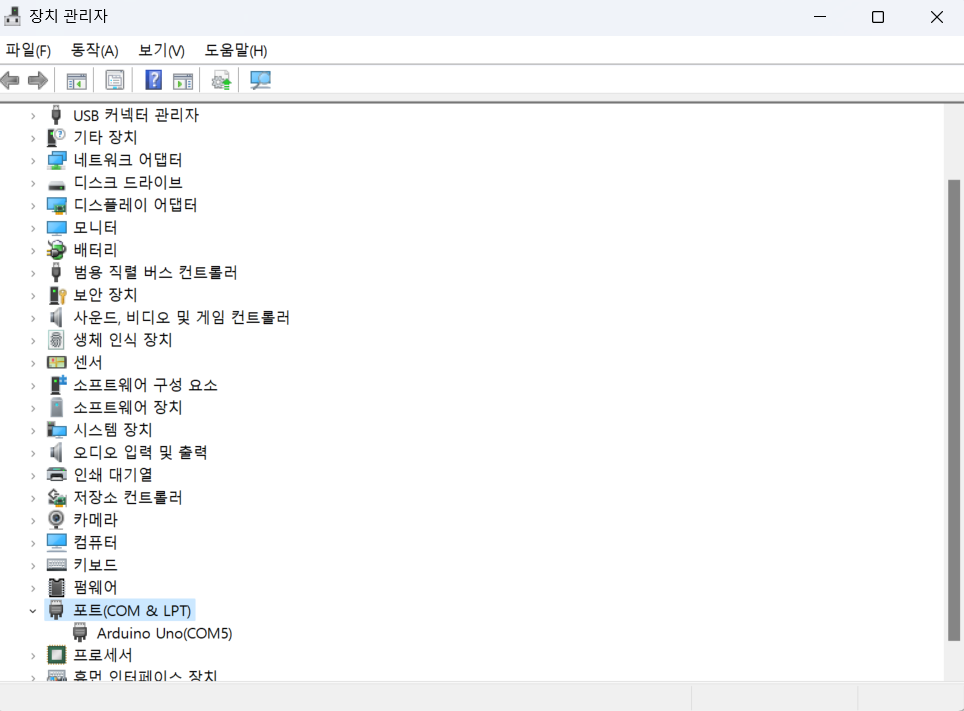
**3. 사용자 메뉴얼**

**3.1 com번호 입력**

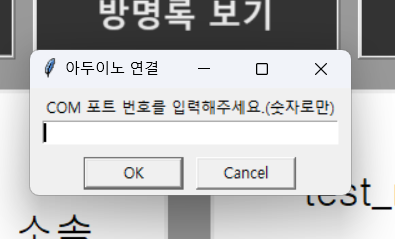
- 장치관리자를 통해 아두이노 uno의 com포트 번호를 확인.

- 프로그램 실행 시 com포트 번호 입력

<그림1 3.1 장치관리자>



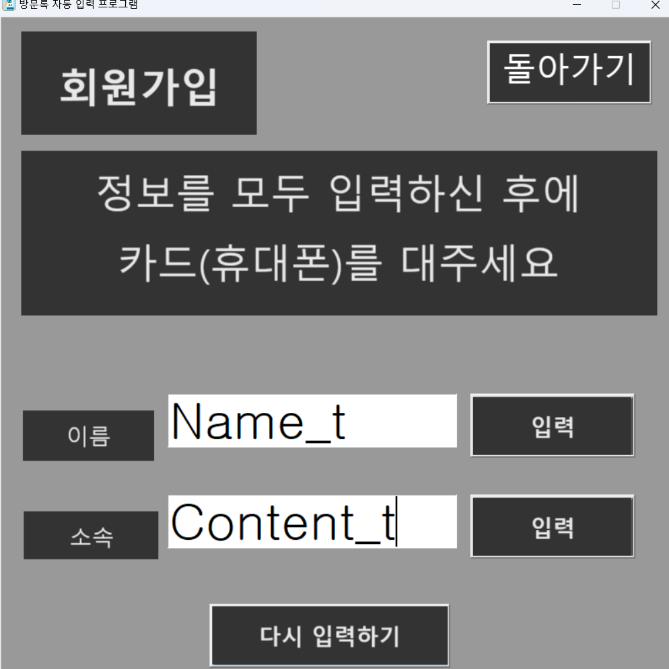
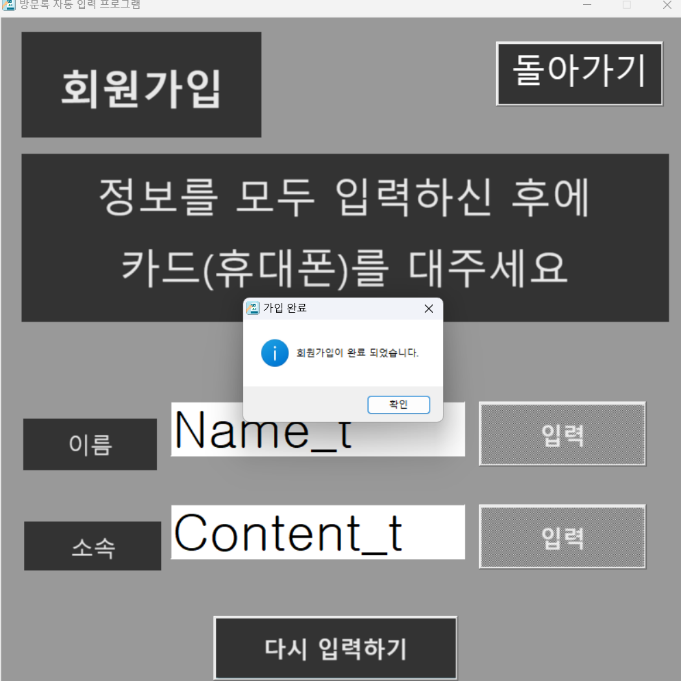
<그림2 3.1 프로그램 시작 com 포트 번호 입력>



**3.2 회원가입**

- 회원가입 화면에서 성명과 소속을 입력 후 '입력' 버튼을 누른다. 이후 자신의 NFC카드를 인식하면 회원가입이 완료됨

<그림1,2 3.2 회원가입 화면>

**3.2.1 회원가입 이후 화면 갱신**

- 회원가입 직후 메인화면의 왼쪽 하단에 회원가입이 되었음을 출력

<그림 3.2..1 회원가입 직후 메인화면>



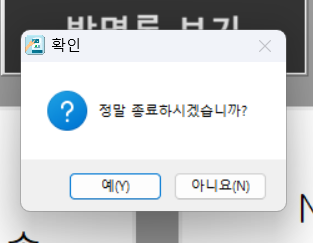
**3.3 메인화면**

- 메인화면은 '회원가입', '방명록보기', '종료' 버튼으로 구성됨.

- 회원가입이된 NFC가 인식될 때 오른쪽 하단에 방문 여부를 출력

- 종료버튼을 누르면 messagebox로 종료여부를 확인 후 종료

<그림1,2 3.3 NFC인식, 종료버튼>

**3.4 방명록 보기 화면**

- '돌아가기' 버튼, 페이지를 넘기는 버튼, 정렬 버튼으로 구성

- 현재 db에 저장된 데이터를 가져와 방명록 화면에 출력함

- 화면 최대 15개의 데이터를 출력해주며 그 이상일 경우 다음 화면으로 넘어갈 수 있는 ' 화살표' 버튼이 활성화됨

- '날짜', '시간', '이름', '소속'으로 정렬할 수 있음

-  

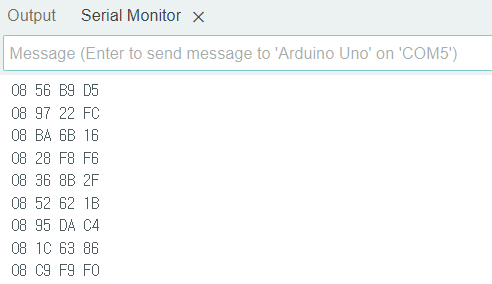
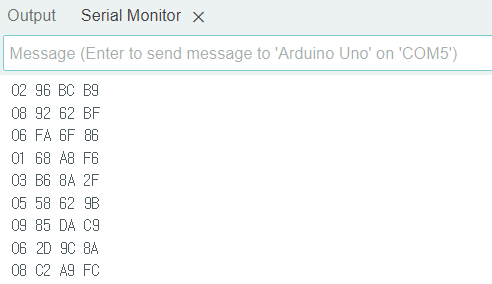
**4. 문제점**

**4.1 인지된 문제점**

- 아두이노 키트의 rfid 태그와 카드는 고유한 uid를 가지나 안드로이드 스마트폰의 nfc의 경우 센서에 인식될때마다 uid가 바뀜

- 안드로이드의 경우 처음 두자리가 동일하지만 아이폰의 경우 처음 두자리 조차 다름

<그림1,2 4.1 안드로이드 NFC 인식, 아이폰의 NFC 인식>

**4.2 문제점의 대처**

- 현 상태에서 마땅한 해결책 모색을 못함

- 'NFC가 UID가 고유하다'라는 가정에서 프로그램을 제작했기 때문에 UID를 기준으로 사용자를 구분하는 프로그램의 제작은 불가.

- '성명'으로 구분하는 방명록 프로그램으로 바꿀 수 있으나, '아두이노를 활용하는 프로젝트' 라는 취지에서 벗어나기에 불가

**5. 프로젝트 진행**

**5.1 프로젝트 진행 방식**

- **주기적인 모임:** 매주 1회 모임을 가지며 현재 진행 척도 공유 및 피드백을 주고받음.

- **메신저를 통한 실시간 소통:** 프로젝트 팀원들 간에 메신저를 활용하여 프로젝트의 진행 상황을 실시간으로 공유하고 빠르게 피드백을 주고받음

**5.2 역할분담**

1. 정기주

- 주요 역할: 아두이노와 파이썬, 그리고 데이터베이스의 연결 담당

- 조장: 팀 내 리더로서 팀원들을 이끄는 역할

- 문서 작성: 개발 계획서, 최종 보고서, 회의록, 연구 노트 작성

- 발표 담당: 개발 계획서 및 최종 결과물에 대한 발표

2. 금명섭

- 주요 역할: 아두이노 장치 개발 및 코드작성

- 발표 담당: 아두이노 관련 내용에 대한 주차별 발표

- 문서 작성: 연구 노트 작성

- 외형 담당: 프로젝트 결과물의 외형 디자인 담당

3. 전호준

- 주요 역할: 파이썬 GUI 관련 코드작성

- 발표 담당: GUI 관련 주차에 대한 바표

- 문서 작성: 연구 노트 작성

- 외형 담당: 프로젝트 결과물의 외형 디자인 담당