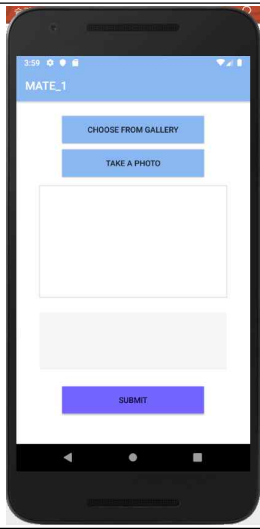


# 공개 소프트웨어 프로젝트 최종보고서

과제 수행원 현황						
수행 학기	<input checked="" type="checkbox"/> 2019년 3월~2019년 6월 <input type="checkbox"/> 2019년 9월~2019년 12월					
프로젝트명	OCR-based Illegal Advertising Block					
팀명	MATE					
	학과	학번	성명	성별	연락처	E-mail
팀장	컴퓨터공학과	2017112064	이주원	여	010.5462.3191	2017112064@dongguk.edu
팀원	컴퓨터공학과	2015112162	신유경	여	010.2790.6503	roodwl@naver.com
	컴퓨터공학과	2017112078	정지연	여	010.7303.6396	jjy1000@dongguk.edu
	컴퓨터공학과	2017112084	신성현	여	010.6555.3285	xltmqp32@dongguk.edu
지도교수	교과목명	공개SW프로젝트_02				
	소속	<input checked="" type="checkbox"/> 컴퓨터공학전공 <input type="checkbox"/> 정보통신공학전공 <input type="checkbox"/> 멀티미디어공학전공 <input type="checkbox"/> 융합소프트웨어연계전공				
	성명	손윤식 교수님				

# Key Words	OCR	RPA	AUTO CALL	JOIN MEMBERSHIP	DATABASE
1.개발동기/목적/필요성 및 개발 목표	<p><b>* 개발동기 및 목적:</b></p> <p>서울시는 성매매를 근절시키기 위해 그간 매수자를 처벌하고 현장 단속을 하여 성매매 업자를 찾아내는 방법을 쓰다가 한계를 느끼고 성매매업자의 전화를 차단하는 방식을 도입하였다. 그러나 전단지에 적힌 전화번호의 사용을 정지시키기 위해서는 통신회사들과 협력해야 하는데 실제 정지까지 일주일 정도 걸리다 보니 그사이에 업자는 새로운 대포폰으로 갈아타 버려 실효성이 없었다. 그래서 아예 성매매 업자가 영업할 수 없도록 일종의 "영업 방해"를 하는 시스템을 개발하였다. 전단지에 적힌 업자의 전화번호로 3초에 한 번씩 자동으로 시스템이 전화하는 방식이다. DDoS 식의 통화 공격으로 계속 통화 중에 걸려 성매수자로부터 연락을 못 받게 한다.</p> <p>이러한 기존의 방법에는 자원봉사자들이 전단지를 줍고, 이를 경찰서에 가져다주면 그 수거된 전단지의 번호를 수사관이 직접 프로그램에 입력해야 하는 "단순노동"이 있다는 것을 파악하였다. 이에 우리 팀은 수거된 전단지를 보고 직접 번호를 프로그램에 입력하는 것이 아니라 전단지 사진만 찍으면 바로 전화번호가 데이터베이스에 올라가는 RPA방식을 구현해보자는 생각을 하게 되었다. 또한, 기존의 대포킬러 대신 다수의 인원이 직접 전화를 걸어 불법 전단지의 번호를 무력화시키는 애플리케이션을 구현하고자 하였다. 물론 통신사와의 협약을 통한 기존 대포킬러 프로그램이 더</p>				

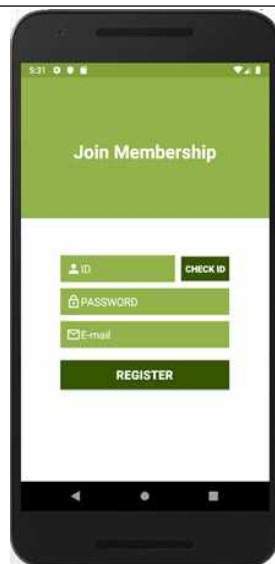
	<p>효율적이고 직관적인 방법일 수 있지만, 불법을 막는 일에 사람들의 좀 더 적극적인 참여를 요구하고 싶다는 생각이 들었다.</p> <p>따라서 프로젝트의 주제를 “OCR-based Illegal Advertising Block”으로 정하여 사회 문제에 사람들이 좀 더 경각심을 가지게끔 유도하고, 이것이 효율적인 방식으로 처리될 수 있도록 하는 애플리케이션을 제작하기로 하였다.</p> <p>해당 프로젝트의 목적은 다음과 같다.  “ 생산성이 저하되는 단순노동 일을 없애고  불법을 막는 일에 사람들을 적극적으로 참여시킨다. ”</p> <p><b>* 개발목표:</b>  프로젝트는 총 두 개의 애플리케이션을 제작한다.  첫 번째 애플리케이션은 불법 전단지를 사진 찍어 정규표현식을 이용해 전단지로부터 휴대폰 번호를 추출하여 데이터베이스에 올리는 역할을 한다. 이 애플리케이션의 사용자로는 경찰이 될 수도, 인증된 자원봉사자 일 수도 있다. 사진으로부터 추출된 번호가 전송 버튼과 함께 바로 데이터베이스에 전송되므로 충분한 교육을 받은 사람이 쓸 수 있다.  두 번째 애플리케이션은 로그인 후 데이터베이스로부터 전화번호를 불러와서 자동으로 전화를 걸고 이용시간을 체크하는 역할을 한다. 이 애플리케이션은 다수의 자원봉사자가 쓸 수 있으며, 이때 자원봉사자들은 첫 번째 애플리케이션을 사용하는 사람과 같을 수도, 다를 수도 있다. 해당 자원봉사자들은 이 애플리케이션으로 전화를 건 시간에 대한 봉사시간으로의 보상이 주어지게 된다. 전화를 거는 자원봉사자들은 자신들의 발신 번호가 수신자에게 알려지는 것을 꺼릴 수 있으므로 "발신전화표시제한"으로 걸리도록 한다.</p>
<p><b>2.최종 결과물 소개</b></p>	<p>GitHub에 올려진 두 개의 애플리케이션을 Android Studio를 이용해 핸드폰으로 다운받아 이용할 수 있다.</p> <p>1. 사진을 찍어 데이터베이스 전화번호를 전송하는 앱  애플리케이션 이름 : MATE_1 (github의 Android-OCRSample-master)</p> <div data-bbox="419 1462 1377 1989"> <div data-bbox="419 1462 898 1989">  </div> <div data-bbox="898 1462 1377 1989"> <p>앱의 첫 실행화면이다.</p> <p>CHOOSE FROM GALLERY를 클릭해 사진 갤러리에서 사진을 선택할 수 있고 TAKE A PHOTO를 눌러 바로 사진을 찍을 수 있다.</p> </div> </div>

	<p>TAKE A PHOTO를 클릭해 전단지의 사진을 찍는다.</p>
	<p>찍은 전단지의 번호와 인식된 번호를 비교할 수 있다.</p> <p>사진이 흔들리거나 제대로 인식이 안 된 경우 다시 사진을 찍을 수 있고 전단지의 번호와 인식된 번호가 일치하면 SUBMIT 버튼을 클릭해 데이터베이스에 전화번호를 전송한다.</p>
	<p>데이터베이스에 번호가 전송된 후 전송 성공을 알리는 success가 나타난다.</p>

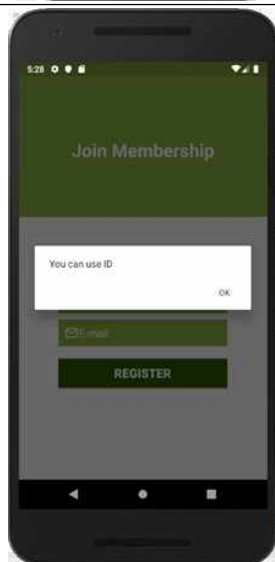
```
mysql> select * from phone;
+----+-----+
| id | phonenumber |
+----+-----+
| 1  | 01027906503 |
| 3  | 01054623191 |
| 10 | 01054623191 |
| 13 | 01027906503 |
| 15 | 01027906503 |
| 16 | 01022258841 |
+----+-----+
6 rows in set (0.00 sec)
```

데이터베이스에 전화번호가 들어왔음을 볼 수 있다.

2. 데이터베이스에서 번호를 불러와 전화를 거는 앱  
애플리케이션 이름 : MATE\_2 (github의 Timer-master)

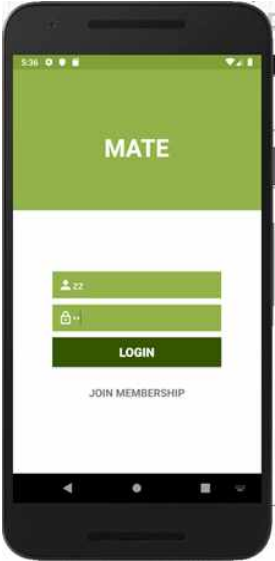

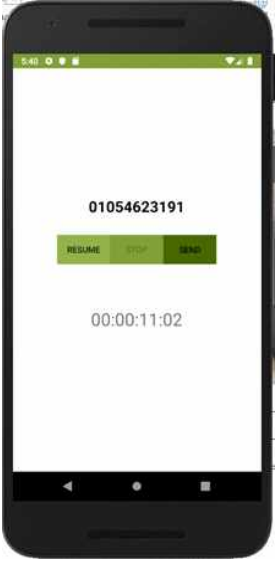


아이디, 비밀번호, 이메일을 이용해 회원가입을 진행한다.



아이디를 입력한 후 CHECK ID를 클릭하면 사용 가능한 아이디인지 확인한다. 사용 가능한 아이디면 사용 가능함을 알리고 아이디 창이 회색으로 바뀐다.



		가입한 아이디와 비밀번호를 이용해 로그인한다.
		로그인 시 전화가 걸 수 있는 화면으로 넘어온다.
		START 버튼 클릭 시 데이터베이스에서 전화번호를 랜덤하게 읽어와 전화를 걸기 시작하고 시간 카운트를 시작한다.



전화를 걸자 발신 번호 표시 제한으로 전화가 가는 것을 확인할 수 있다.  
전화 종료 버튼 클릭 시 전화가 종료되고 전화가 종료되면 다시 전화를 걸기 시작한다.



SEND 버튼 클릭 시 시간이 데이터베이스에 전송되고 0으로 초기화된다.

```
mysql> select * from USER;
+----+-----+-----+-----+
| userID | userPassword | userEmail | userTime |
+----+-----+-----+-----+
| aa | aa | aaaaa | 00:00:58 |
| bb | bb | bb@naver.com | 00:00:23 |
| cc | cc | ccc | 00:00:45 |
| dd | dd | dd@gmail.com | 00:00:00 |
| ee | ee | eeeee | 00:00:19 |
| jj | jj | jj@dongguk.edu | 00:01:01 |
| rr | rr | rr | 00:00:51 |
| zz | zz | zz | 00:00:00 |
+----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> select * from USER;
+----+-----+-----+-----+
| userID | userPassword | userEmail | userTime |
+----+-----+-----+-----+
| aa | aa | aaaaa | 00:00:58 |
| bb | bb | bb@naver.com | 00:00:23 |
| cc | cc | ccc | 00:00:45 |
| dd | dd | dd@gmail.com | 00:00:00 |
| ee | ee | eeeee | 00:00:19 |
| jj | jj | jj@dongguk.edu | 00:01:01 |
| rr | rr | rr | 00:00:51 |
| zz | zz | zz | 00:00:11 |
+----+-----+-----+-----+
8 rows in set (0.00 sec)
```

zz로 로그인해 11초간 이용한 후  
SEND 버튼을 누르자 데이터베이스  
의 이용시간이 11초 증가했음을 볼  
수 있다.

### 3.프로젝트 추진 내용

#### 1. 프로젝트 진행과정

내용	3월			4월				5월				6월			
	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주
주제 선정	→														
Open Source 분석															
Android Studio 공부															
SERVER 공부															
SERVER 구현															
SERVER 변경															
DATABASE 구현															
앱 (MATE.1) 구현															
앱 (MATE.2) 구현															
앱 수정															
UI 구현															
최종발표															

- **주제 선정** : 지금의 'OCR 기반 불법 광고 차단'주제를 선정하기까지 면담 및 주제 변경의 반복이 있었다.
- **Open Source 분석** : 사진을 찍어 OCR 인식을 수행하는 기존의 오픈소스 안드로이드 앱의 소스 분석을 수행했다.
- **Android Studio 및 SERVER 공부** : 안드로이드 앱을 만들어본 경험이 없고 서버도

제대로 배운 적이 없어 오픈소스 분석과 함께 안드로이드 스튜디오와 서버에 관한 공부를 진행했다.

· **SERVER 구현 및 변경** : 처음 server는 네이버 클라우드를 이용해 구현했으나 요금 부과 문제로 서버를 변경해 학교 연구실의 서버를 이용했다. 하지만 서버에 다수가 접근하자 차단되는 문제가 발생하여 다시 한번 서버를 변경하게 되었다. 최종 서버는 Ubuntu를 이용해 구현했다.

· **DATABASE 구현** : 데이터베이스는 mysql을 이용해 구현했다. 하나의 데이터베이스에 회원 테이블(USER)과 전단지 전화번호 저장 테이블(PHONE)을 나눠 저장한다.

· **앱(MATE\_1) 구현** : MATE\_1은 OCR 인식을 수행하고 데이터베이스에 번호를 올리는 앱으로 Android Studio를 이용해 제작했으며 기존의 OCR 인식 앱을 수정해 번호만 인식하게 하고 인식한 번호와 사진을 비교한 후 데이터베이스에 올리도록 확장했다. 데이터베이스에 번호를 올리기 위해 php를 이용해 통신한다.

· **앱(MATE\_2) 구현** : MATE\_2는 데이터베이스에서 번호를 읽어와 전화를 거는 앱으로 Android Studio를 이용했으며 php를 이용해 서버의 데이터베이스와 통신한다. 기존의 시간만 카운트하는 타이머 앱을 수정해 로그인 기능과 전화번호를 랜덤하게 읽어 온 후 자동으로 전화를 거는 기능을 추가하고 시간을 카운트해 데이터베이스에 전송하게 제작했다.

· **앱 수정** : 처음 제작한 MATE\_1은 전화번호를 인식한 후 찍은 사진 속 전화번호와 비교할 수 없었으나 수정을 통해 찍은 사진 속 전화번호와 비교할 수 있게 변경했다. 이 외에 앱에서 매끄럽지 못한 부분을 수정해 완성도를 높였다.

· **UI 구현** : 이전까지 앱은 기능 구현에 중심을 두고 제작해 앱 아이콘부터 앱 내부까지 디자인을 변경했다.

· **최종 발표** : 진행한 프로젝트에 대한 최종 발표를 진행했다.

## 2. 프로젝트 구현과정

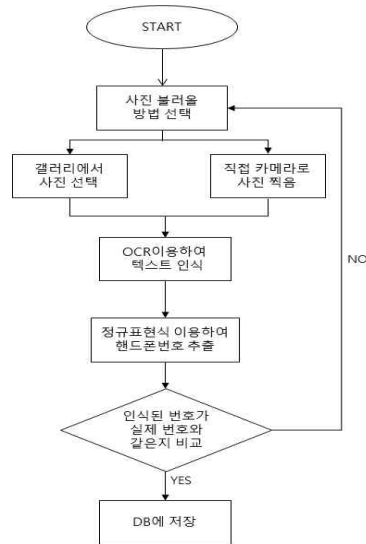


그림 1 프로그램 개발 환경

이 프로젝트의 안드로이드 애플리케이션 개발은 Android Studio상에서 구현하였고, 서버는 Ubuntu에 MySQL을 설치하여 환경을 구축하였다. 필요한 php파일의 생성 및 관리리는 putty를 통해 이루어졌다. 프로젝트는 다음과 같이 2개의 안드로이드 어플리케이션으로 구성된다.



## 2.1. MATE\_1 (사진을 찍어 데이터베이스에 전화번호를 전송하는 앱)



id	phonenum
----	----------

표1 phone 테이블 구조

그림2 MATE\_1 순서도

### 2.1.1. 구현 내용

사진을 찍어 데이터베이스에 업로드하면 전화를 계속하게 되므로 경찰, 신원이 보장된 자원봉사자만 이용할 수 있다. 사용자가 앱으로 사진을 찍으면 OCR로 텍스트를 인식하여 전화번호만을 인식한다. 인식된 전화번호가 실제 사진의 번호와 일치하는 것이 확인되면 사용자는 'submit'버튼을 클릭하여 이 번호가 데이터베이스에 저장되도록 한다.

### 2.1.2. 구현 방법

#### ● 선택한 사진에서 전화번호 추출

- 상세 설명: Google에서 제공하는 API인 TextRecognizer를 이용하여 OCR 기술로 사진에 있는 모든 텍스트를 인식한다. 그 후, 정규표현식 : "01(?:0|1).( \w \w {3 }| \w \w {4 }).( \w \w {4 })"을 이용하여 필요한 정보인 핸드폰 전화번호만을 추출하여 화면에 보여준다.

#### ● 데이터베이스에 저장

- 상세 설명: 웹 서버에 insertPhone.php파일을 생성하고 안드로이드 앱에서 php파일을 통해서 MySQL에 접속하는 class인 insertData를 생성한다. 앱에서 POST 방식으로 php 코드에 데이터를 전달하면 php코드에서 MySQL에 접속하여 추출한 전화번호를 저장한다.

## 2.2. MATE\_2 (데이터베이스에서 번호를 불러와 전화를 거는 앱)

### 2.2.1 구현 내용

자동 전화를 구현하는 앱으로 자원봉사자의 참여가 필요하기 때문에 회원가입 후 이용할 수 있다. 회원가입 후 성공적으로 앱에 로그인하면 자동전화를 걸 수 있는 화면이 나타난다. START 버튼을 클릭하면 자동전화가 시작함과 동시에 앱의 타이머가 동작한다. 자원봉사자의 사용 유도를 위해 이 시간을 USER 테이블에 저장하고, 일정 시간 이상 사용 시 봉사자에게 봉사시간을 부여한다.

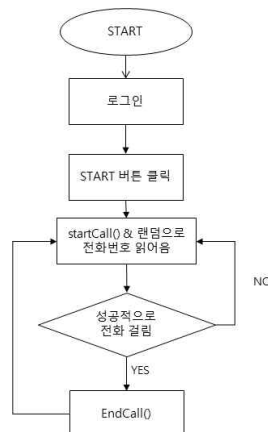


그림 3 MATE\_2 순서도

UserID	UserPassword	UserEmail	UserTime
--------	--------------	-----------	----------

### 2.2.2 구현 방법

표2 USER 테이블 구조

#### ● 로그인 및 회원가입

##### (1) 회원가입

- 상세 설명 : 필요한 정보인 ID, Email, 비밀번호를 입력한다. 이때 checkID를 통하여 UserValidate.php에 접속하여 USER 테이블에 입력한 ID가 있는지 검사한다. 응답이 false인 경우 중복된 ID이므로 ID를 다시 입력해야 한다. 모든 항목이 정상적으로 입력되면 앱에서 POST 방식으로 UserRegister.php에 데이터를 저장하고 php코드에서 USER 테이블에 회원 데이터를 저장한다. 회원가입이 끝나면 로그인 페이지로 이동한다.

##### (2) 로그인

- 상세 설명 : 사용자가 ID와 비밀번호를 입력하면 앱에서 UserLogin.php에 POST 방식으로 데이터를 전달한다. php코드에서 입력한 데이터와 일치하는 회원 데이터가 USER테이블에 존재하는지 검사하여 응답이 success이면 다음 화면으로 넘어가고 아닐 경우 로그인을 다시 시도한다.

#### ● 데이터베이스에서 전화번호 선택

- 상세 설명 : phone테이블의 전화번호를 가져오기 위해 GettingPHP라는 새로운 class를 만들었다. 이 class를 통해 서버의 phone.php에 접속하고, php코드는 테이블에 저장된 전화번호 데이터들을 전달해준다. 그 후 JSON 배열에서 랜덤으로 하나의 JSONObject를 선언하고 string으로 변환하여 전화를 걸 번호 변수에 저장한다.

- 자동 전화

- 상세 설명: 사용자가 로그인에 성공하고, START를 누르면 전화발신과 타이머가 시작된다. 반복적인 전화를 위해 Handler를 이용하여 메시지를 보내는 방식을 사용하였다. 받은 메시지가 'startCall'인 경우 전화를 거는 함수인 startCallPhone()을 실행하고, 'stopCall'인 경우 전화를 받는 함수인 stopCallPhone()을 실행한다. startCallPhone()에서는 GettingPHP class를 실행시켜 전화번호를 랜덤으로 가져오고, 이 전화번호에 \*23#을 추가하여 수신자에게 발신자의 번호가 보이지 않도록 하였다. 그리고 intent객체를 생성하여 ACTION\_CALL 옵션을 선언하고, 위에서 받은 전화번호 데이터를 넣어준다. startActivity(intent)를 통해 전화를 걸게 된다. 이때 TelephonyManager를 통해 실시간으로 기기의 상태를 받아올 수 있는데, CALL\_STATE\_IDLE(0)인 경우 똑같이 전화를 걸고, CALL\_STATE\_RINGING(2)인 경우 Handler에 메시지를 'stopCall'메세지를 보낸다. stopCallPhone()에서도 실시간으로 기기의 상태를 받아와서 CALL\_STATE\_RINGING인 경우 ITelephony의 endCall()을 이용하여 전화를 끊고, Handler에 'startCall'메세지를 보낸다. endCall()이 시스템 권한 문제로 정상적으로 작동되지 않아서 해결 사항으로 남겨두었지만 기기의 상태정보를 계속 받아오기 때문에 기기가 전화를 걸고 있지 않은 상태면 계속해서 전화를 걸 수 있다. 이 자동전화 프로세스는 사용자가 STOP 버튼을 누르기 전까지 반복된다.

- 데이터베이스에 저장

- 상세 설명: 사용자가 SEND 버튼을 클릭하면 앱에서 UserTimer.php에 POST 방식으로 사용자의 ID와 이용시간을 전달한다. php코드에서는 해당하는 USER테이블에서 ID와 일치하는 회원을 찾아 기존에 저장된 시간 값에 이용시간을 추가하여 업데이트한다.

### 2.3. 기존 대포킬러와의 차이점

기존 대포킬러는 자원봉사자가 수거한 전단지를 경찰서에 서에 전달하면 수사관이 전달받은 번호를 프로그램에 하나하나 직접 입력해야 하는 번거로움이 있었다. 이 불편함을 해소하기 위해 RPA 개념을 적용하여 이 과정을 자동화시켰다. 이 프로젝트에서 자원봉사자가 전단지의 사진을 찍으면 OCR로 사진에서 번호인 텍스트를 인식하여 자동으로 데이터베이스에 전달된다. 또한 대포킬러는 프로그램 하나로 3초마다 한 번씩 전화를 걸지만 이 프로젝트는 보다 많은 자원봉사자들의 참여를 유도하고자 새로운 안드로이드 앱을 제작하였다. 이 앱은 데이터베이스에 저장된 전화번호를 랜덤으로 하나씩 불러오고, 사용자가 시작 버튼을 누르면 자동으로 전화를 건다. 대포킬러처럼 자동으로 전화를 끊고 거는 것이 아니라 앱이 설치된 장치의 상태 정보를 실시간으로 받아와서 사용자나 수신자가 전화를 끊었을 때 다시 전화를 거는 방식으로 디도스 공격을 구현하였다.

<p><b>4.기대효과</b></p>	<p>대포킬러는 본래 대포폰을 대상으로 작동하는 프로그램으로 시스템에 등록된 전화번호로 3초마다 전화를 걸어 전화번호를 사용하게 하지 못하도록 하는 프로그램이다. 길거리에 붙은 성매매 전단지나 알선 사이트 광고 속 전화번호를 수집해 프로그램을 가동시켜 성매매 알선자와 구매자 간의 접촉을 원천 봉쇄한다.</p> <p>이는 컴퓨터 해커들이 특정 서버나 홈페이지에 과도한 트래픽을 흘려보내 서버가 먹통 되게 하는 디도스(DDoS-분산서비스거부공격) 공격과 비슷하다. ‘이이제이(以夷制夷)’ 오랑캐로 오랑캐를 막는다는 고사 성어처럼 해킹 범죄의 수법으로 전화폭탄을 날려 성매매를 막는 아이디어이다.</p> <p>이 시스템은 현재 민생사법경찰단(민사경)과 자치구, 자원봉사단의 협업으로 100여명의 자원봉사자가 거리 곳곳의 성매매 전단을 수거해 민사경에 전달하면 민사경은 번호를 대포킬러에 입력한다.</p> <p>우리가 만든 앱은 자원봉사자의 핸드폰 여러 대로 랜덤하게 전화를 걸기 때문에 성매매 업자는 수요자를 구분하기가 어렵다. 즉, 업자가 해당 번호를 차단해도 다른 번호로 전화를 걸어 수요자와의 통화연결 불능을 유도해 사실상 업자는 불법영업을 영위하기 힘든 상황이 된다. 또한, 한 번 등록된 전단지 전화번호는 번호가 사라지기 전까지 지속해서 전화 공격을 하므로 공격을 벗어날 수 없다.</p> <p>이러한 불법 광고 전화 차단시스템을 통해 무차별적인 불법광고물 배포 행위를 차단할 수 있고, 불법 광고업자의 광고행위를 차단해 피해 예방과 유해환경 감소에 효과가 있을 것으로 기대한다. 또한 이런 아이디어는 불법 전단지만이 아니라 보이сп이스 같은 각종 사기 범죄 예방에도 응용이 가능할 것으로 보인다.</p> <p>그뿐 아니라 전화를 거는 것에 일종의 우선순위를 두어, 중요도에 따라 번호 차단 강도에 차이를 둔다면 더 효과적으로 불법 전단지 예방을 할 수 있을 것이라 생각한다.</p>
<p><b>5.구현과정에서의 문제점 및 해결과정</b></p>	<p>* 구현과정에서 있었던 문제점 및 해결과정</p> <p><b>1. OCR 기술 정확도 문제</b></p> <div data-bbox="493 1386 1291 1516" data-label="Diagram"> <pre> graph LR     A[PDF Document] --&gt; B[TXT File]     C[Scanned Document with Photo] --&gt; D[TXT File]   </pre> </div> <p>OCR(Optical Character Recognition, 광학 문자 인식)은 인쇄물 또는 사진상의 글자와 이미지를 디지털 데이터로 변환해주는 자동인식기술이다. 우리가 처음에 이 기술을 사용해 사진의 모든 글자를 인식하고자 했을 때는 상대적으로 한글에 대한 인식률이 좋지 않아 난항을 겪었다.</p> <p>그래서 전단지의 모든 글자를 인식하는 것이 아니라 전단지에서 핸드폰 번호만 인식해 오는 것으로 방향을 바꿔 프로젝트를 진행했다. OCR기술은 최근 기계학습 기반의 알고리즘을 활용하면서 인식률이 크게 높아졌고, 딥러닝을 적용해 정확도가 이전보다 향상되었다. 규격 이미지 등에서 OCR 기술을 적용할 시 약 90%의 정확도를 보인다. 따라서 전단지에서 숫자만 인식해 오는 것에는 문제가 생기지 않았고, 더욱 정확하게 번호만 인식해 오기 위해 Java 정규표현식을 사용해 번호의 형식을 가진 숫자만 읽어 오는 것을 추가했다.</p>

## 2. 타겟 대상 찾기

우리 팀의 어플리케이션 두 가지 중에 첫 번째 앱인 전단지의 사진을 찍어 데이터베이스에 올리는 앱의 사용 대상이 경찰인지, 아니면 자원봉사자일지에 대해 의견이 갈렸다. 우리는 교수님과의 면담을 통해 그 부분에는 기술적인 차이가 없으므로 RPA에 초점을 맞추어 자원봉사자가 바로 불법 전단지를 찍고 데이터베이스에 올리는 방법을 채택했다. 대신 사용대상을 경찰이나 엄선된 자원봉사자로 제한을 두어 불법 전단지의 번호가 아닌 다른 번호가 데이터베이스에 올라가지 않도록 했다.

두 번째 앱은 다수의 사용자가 번호에 전화를 거는 방식이기 때문에 많은 자원봉사자가 필요했다. 따라서 자원봉사자의 사용 유도를 위해 앱에 이용시간을 나타내는 타이머를 추가했고, 일정 시간 이상 사용 시 봉사시간을 부여하게 했다.

## 3. 서버 문제

처음에 사용했던 서버와 데이터베이스는 네이버 클라우드를 이용해서 CentOS와 MySQL을 이용했다. 하지만 예상치 못한 청구 금액이 발생했고, 서버와 데이터베이스를 모두 옮겨야 하난 상황이 왔다. 결국, 랩실에서 가상환경을 통해 웹 서버를 만들 수 있는 우분투 환경을 받았고, Ubuntu에 apache2 웹 서버와 Mysql을 설치해 필요한 새로운 환경을 모두 마련하였다.

그러나 그 후 하지만 서버에 다수가 접근하자 정보처에서 서버 접속을 차단하는 문제가 발생하여 다시 한번 서버를 변경하게 되었다. PHP파일을 사용하였기 때문에 WAS를 이용하여 이를 JSON으로 바꾸었으며 Ubuntu 운영체제 안에 mysql로 데이터베이스를 생성하였다.

## 4. 데이터베이스에서 번호를 가져오는 과정

두 번째 앱에서는 데이터베이스에서 번호를 가져와 전화를 걸어주는데, 이 때 번호를 어떻게 가져올 것인가에 대해 여러 가지 의견이 나왔다. 데이터베이스로부터 가져오는 번호를 랜덤으로 무작정 가져올지, 서버에서 스케줄링해서 가져올지, 아니면 그 번호의 중요도에 따라 판단하게 할지를 고민해보았다. 그 결과, 어차피 자원봉사자들이 찍어서 올리는 번호가 겹칠 수 있고, 데이터베이스에서 중복 제거를 하지 않을 것이므로 데이터베이스에 겹치는 번호가 많다면 그만큼 랜덤으로 가지고 올 때 사진을 많이 찍힌 번호가 선택될 확률이 높아지므로 단순히 랜덤으로 번호를 가져오게 해도 된다고 생각해 그 방법을 선택했다.

## 5. 자동 전화 문제

데이터베이스로부터 랜덤으로 번호를 가져와 자동으로 전화를 걸어주는 두 번째 어플리케이션에서 자동으로 전화는 걸리지만, 그 전화를 끊는 것에서 문제가 발생하여 해결 과정에서 난항을 겪었다.

TelephonyManager를 이용하여 실시간 device의 상태를 받아오고 php를 통해 랜덤으로 전화번호를 추출하였다. 그리고 handler를 이용하여 반복적으로 자동 전화가 시작되도록 하였다. endcall()을 이용하여 전화 끊기를 시도해보았지만, 우리가 진행하는 application의 버전에서는 permission이 허용되지 않아 제대로 작동하지 않았다.

그래서 thread를 이용하여 자동으로 전화가 걸리는 시간을 조절하고, 전화를 하는 것은 intent을 사용하였으며 device 상태에 따라 전화를 걸고 끊고를 반복적으로 구현

	<p>하는 데에 handler를 이용했다. 전화를 걸 애플리케이션에서 자원봉사자가 START버튼을 누르면 데이터베이스에 있는 전화번호를 랜덤으로 하나씩 전달받아 3초마다 전화를 걸고, 자원봉사자가 애플리케이션의 STOP버튼을 누르기 전까지는 자원봉사자가 전화를 끊거나 수신자가 통화를 거절해도 계속 전화를 걸어 수신자의 전화번호를 무력화시키게 했다.</p> <p>하지만 아직 통화 종료나 거절 버튼을 누르기 전에는 전화가 끊기지 않는 한계가 있다. 이는 해결 사항으로 남겨두기로 한다.</p>
<p><b>6.참고문헌</b></p>	<p>■ 참고사이트</p> <p>▶ 대포킬러</p> <p><a href="https://news.sbs.co.kr/news/endPage.do?news_id=N1005201237">https://news.sbs.co.kr/news/endPage.do?news_id=N1005201237</a></p> <p><a href="http://biz.heraldcorp.com/view.php?ud=20180706000687">http://biz.heraldcorp.com/view.php?ud=20180706000687</a></p> <p>▶ Opensource</p> <p><a href="https://github.com/komamitsu/Android-OCRSample">https://github.com/komamitsu/Android-OCRSample</a></p> <p><a href="https://github.com/KayReid/Timer">https://github.com/KayReid/Timer</a></p> <p><a href="https://github.com/GaKaRi/gakari_android/tree/master/Registration_v4">https://github.com/GaKaRi/gakari_android/tree/master/Registration_v4</a></p>