

소프트웨어 포트폴리오

프로젝트 개요

-명칭: Blood Community(헌혈 커뮤니티)

-개발 인원: 3 (개발자 3명)

-맡은 부분: 레이아웃 1/2, 회원가입/로그인 기능, 회원정보 수정, 바코드 인식/저장 기능, 웹 크롤링 기능, 최종 디자인

-참여 기간: 2019.09.05~2019.12.23

-개발 환경: 안드로이드 스튜디오(JAVA), API 이용

-프로젝트 링크: <https://github.com/psyoun99/bloodcommunity.git>

소개

어플에 회원가입과 로그인을 하면 헌혈증과 관련한 글을 쓸 수 있고, 작성 리스트를 길게 누르면 글 작성자와의 채팅방이 개설되어 채팅을 할 수 있습니다. 헌혈 후 받은 사은품을 스마트폰으로 관리할 수 있도록 바코드 인식, 저장 기능을 구현하였습니다. 대한적십자사 웹페이지에서 업데이트되는 정보를 볼 수 있습니다.

프로젝트 목적

헌혈증을 기부하고 싶다는 사람과 헌혈증을 요청하는 사람들이 도움을 구할 곳이 마련되어 있지 않아 여러 사이트에 자신의 요청사항을 작성하는 것을 보고 헌혈증을 기부하고 요청할 수 있는 전용 커뮤니티를 만들어야겠다는 생각이 들어 프로젝트를 만들게 되었습니다.

사용기술

Firebase의 실시간 데이터베이스를 사용하여 회원가입과 로그인, 회원정보 관리 기능을 구현하였습니다. 각 ID의 자식으로 ID 사용자의 정보를 저장했습니다.

바코드 인식 기능을 구현하기 위해 Zxing API를 사용하였습니다. Zxing 라이브러리는 바코드를 스캔하여 코드와 이미지를 인식할 수 있고, 바코드 이미지를 커스텀으로 생성해 바코드를 추후에 사용할 수 있습니다.

대한적십자사 홈페이지의 헌혈 관련 이벤트 정보 이미지를 크롤링하기 위해 Jsoup 라이브러리를 사용하였습니다. Jsoup 라이브러리를 통해 JSON을 만들고 파싱을 통해 크롤링했습니다.

개발 상세설명

```
public View onCreateView(@NonNull LayoutInflater inflater,
                        ViewGroup container, Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_gallery, container, attachToRoot: false);
    mId = getActivity().getIntent().getExtras().getString(key: "id");
    Log.d(tag: "정보", msg: "barcode -- create" + mId);

    //바코드 Reader 호출
    IntentIntegrator integrator = new IntentIntegrator(getActivity());
    IntentIntegrator integrator = IntentIntegrator.forSupportFragment(this);
    integrator.setBarcodeImageEnabled(true);
    integrator.initiateScan();
    return view;
}
```

```
//스캔 결과값 가져와서 DB 저장
public void onActivityResult(int requestCode, int resultCode,
                            Intent data) {

    Log.d(tag: "정보", msg: "barcode -- dialog show");
    result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    if (result != null) {
        if (result.getContents() == null) {
            Toast.makeText(getContext(), text: "카메라가 취소되었습니다", Toast.LENGTH_SHORT).show();
            return;
        }
        //바코드 이름,유효기간(사용자 직접 입력) -> dialog로 구현
        final Camera_afterdialog dialog = new Camera_afterdialog(getContext(), title: "바코드 이름 저장");
        dialog.setContent(R.layout.camera_after_dialog);
        //dialog size
        ViewGroup.LayoutParams params = dialog.getWindow().getAttributes();
        params.width = ViewGroup.LayoutParams.MATCH_PARENT;
        params.height = ViewGroup.LayoutParams.WRAP_CONTENT;
        dialog.getWindow().setAttributes((android.view.WindowManager.LayoutParams) params);

        dialog.show();
    }
```

```

//dialog 저장버튼 후 DB 저장
dialog.setOnCancelListener((dialogInterface) → {

    //realtime db 인스턴스 생성
    FirebaseDatabase firebaseInstance = FirebaseDatabase.getInstance();
    FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();
    DatabaseReference databaseReference = firebaseInstance.getReference(path: "GifticonInfo");
    //데이터 저장
    // Log.d("정보", dialog.barcode_name);
    GifticonDB GifticonInfo = new GifticonDB(mId, result.getContents(),
        result.getBarcodeImagePath(), dialog.barcode_name, dialog.barcode_due);

    databaseReference.child(result.getContents()).setValue(GifticonInfo);
    Toast.makeText(getApplicationContext(), text: dialog.barcode_name + " 바코드가 저장되었습니다", Toast.LENGTH_LONG).show();
});

```

위의 코드는 바코드를 스캔한 결과를 DB에 저장한 후 dialog 창에 사용자가 바코드의 이름과 유효기간을 작성하여 저장할 수 있도록 하는 부분입니다. Zxing 라이브러리를 사용해 호출한 바코드 리더기의 내용이 null이 아니면 dialog를 선언하여 이름과 유효기간을 입력하도록 한 후 Firebase의 DB에 저장합니다.

```
private void slideshowFunction(View view) {
    web = view.findViewById(R.id.web);
    web.getSettings().setJavaScriptEnabled(true);

    JsoupAsyncTask jsoup = new JsoupAsyncTask();
    jsoup.execute();
}
```

```
private class JsoupAsyncTask extends AsyncTask<Void,Void,Void> {
    @Override
    protected void onPreExecute() { super.onPreExecute(); }

    @Override
    protected Void doInBackground(Void... voids) {
        try {
            Document doc = Jsoup.connect( url: "https://www.bloodinfo.net/event.do?action=detail&eventno=154213&current."
            String title = doc.title();
            Element UriEle=doc.select( cssQuery: "img[alt=전국 헌혈의 집 이벤트 상세정보는 아래 내용을 참고하세요]").first();
            urlpath = UriEle.attr( attributeKey: "src");
        } catch (IOException e) {
        }

        return null;
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        web.loadUrl(urlpath);
        Log.d( tag: "정보", msg: "Open: "+urlpath);
    }
}
```

위의 코드는 대한적십자사 웹사이트의 특정 이미지를 크롤링하는 부분입니다. JsoupAsyncTask 클래스를 선언하여 특정 URL과 연결하고, 웹사이트의 html코드를 이용하여 이미지 경로에 맞는 이미지를 가져온 후 실행하면 원하는 이미지가 크롤링됩니다.

실행결과 이미지



