

소프트웨어 포트폴리오

1. 지능형 나이스 구축 프로젝트

프로젝트 개요

-참여 인원: 2 (개발자 2 명)

-맡은 부분

1. 나이스 웹사이트(유치원 환경관리 메뉴)의 백엔드 개발 담당, 고객 요구사항 반영 작업
2. 나이스 웹사이트(학교행정 교육복지 메뉴) 안정화 작업

-참여 기간: 2023.02.01~2023.05.31 / 2023.06.19~

-개발 환경: Eclipse(JAVA), DB2(DB), Tomcat(서버)

소개

학부모, 학생이나 학교와 관련된 행정 정보를 제공하기 위한 서비스입니다. 저는 유치원 행정정보 서비스를 개발 및 구축했습니다. 고객 편의성에 맞춰 화면을 개발했고, 테스트를 진행하며 오류사항이나 요청사항을 반영해 수정했습니다.

이후 학교 교육복지 서비스의 고객 요청사항과 오류 증가로, 추가 투입되어 고객 요청사항에 대응하고 DB 를 조회하며 테스트를 진행해 웹사이트를 개선하고 있습니다.

사용기술

- ✓ Java Spring
- ✓ eXBuilder(UI 개발 툴)
- ✓ Java
- ✓ SQL(DBeaver)
- ✓ SVN
- ✓ ChangeFlow(형상 관리)

개발내용

Java Spring 프레임워크를 기반으로 유치원 환경관리 메뉴에 해당하는 화면의 백엔드 부분을 주로 개발했습니다. Controller, Service, DAO MVC 구조에 맞게 개발하며 데이터가 제대로 추가, 조회, 삭제되는지 확인했습니다. SQL 쿼리를 작성해 시험용 데이터를 추가하거나, 데이터의 내용을 확인하기

위해 조회하며 개발했습니다. 이후 필요에 따라 분기처리 작업을 통해 데이터가 이상 없이 출력될 수 있도록 하였습니다.

화면 개선을 위해 화면 UI 를 변경하여 가시성과 편리성을 높일 수 있도록 하였습니다. 고객 요청사항에 따라 SQL 을 통해 화면 상에 조회되는 데이터 컬럼을 변경하거나, SQL 을 재확인하며 오류사항을 해결하였습니다.

SVN 을 통해 소스관리를 하고, ChangeFlow 툴을 활용해 개발/운영 서버로 배포하기 위해 승인/합의/반려 기능을 활용했습니다.

또한 교육복지 담당 부서에서 고객의 요청사항, 오류와 관련된 SR 을 처리하며 서비스 안정화 작업을 수행했습니다. 정보가 누락된 경우, 요청사항을 확인한 후 해당 시도의 DB 에서 SQL 쿼리를 작성하여 학생의 현재 데이터를 조회해 문제의 원인을 찾았습니다. 이후 문제상황을 관리시스템에 게시하고 고객과 개발자에 피드백하는 업무를 진행하고 있습니다.

2. 지능형 공무원연금복지 구축 프로젝트

프로젝트 개요

-참여 인원: 5 (개발자 3 명)

-맡은 부분: 모바일 웹 서비스의 프론트엔드 담당(UI 설계/개발), 고객 요구사항 반영 작업

-참여 기간: 2022.01.01~2022.12.30

-개발 환경: Eclipse, Oracle(DB), Tomcat(서버)

소개

모바일 부서에서 공무원 고객이 공무원 연금 관련 업무를 볼 수 있도록 연금복지포털 모바일 웹 서비스를 설계, 개발했습니다. 퇴직급여 조회, 재직정보 확인이나 연금대출 신청 등의 화면 UI 를 개발했습니다. 고객의 요구사항에 따라 개발 후 꾸준히 UI 를 개선했습니다.

사용기술

- ✓ eXBuilder(HTML 기반 UI 개발 툴)
- ✓ Javascript
- ✓ SQL(DBeaver)
- ✓ SVN
- ✓ Jenkins(배포)

개발내용

UI 설계서를 기반으로 exBuilder UI 툴과 Javascript 를 활용해 퇴직급여 예상액 조회, 재직정보 조회 등 연금복지포털의 화면을 개발했습니다. 모바일 화면에 맞게 기존 서비스의 화면 UI 를 새로 설계하며

개발했습니다. 또한 SQL 쿼리를 작성해 DB 의 데이터를 확인하며 툴 상에서 화면의 각 항목에 출력되어야 할 데이터를 매핑하는 작업을 진행했습니다.

SVN 을 통해 동료와 협업하고, 작업한 내용을 확인한 후 Jenkins 를 통해 개발/운영 서버에 배포했습니다.

이후 테스터의 피드백과 고객의 요청사항을 반영하여 개발내용을 수정하였습니다. 웹접근성 기준에 적합하도록 UI 를 개선했고, 화면 상의 데이터가 제대로 출력되지 않는 오류사항을 수정하거나 고객의 요청사항에 따라 특정 버튼을 눌렀을 경우 동적으로 UI 가 나타날 수 있도록 변경하는 등의 작업을 진행했습니다.

3. 헌혈 커뮤니티 모바일 앱

프로젝트 개요

-명칭: Blood Community(헌혈 커뮤니티)

-개발 인원: 3 (개발자 3 명)

-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/psyoung99/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

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

바코드 인식 기능을 구현하기 위해 Zxing API 를 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 JSON 을 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.setContentView(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();
    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 DB 에 저장한 후 dialog 창에 사용자가 바코드의 이름과 유효기간을 작성하여 저장할 수 있도록 하는 부분입니다. Zxing 라이브러리를 사용해 호출한 바코드 리더기의 내용이 null 이 아니면 dialog 를 선언하여 이름과 유효기간을 입력하도록 한 후 Firebase 의 DB 에 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(); }

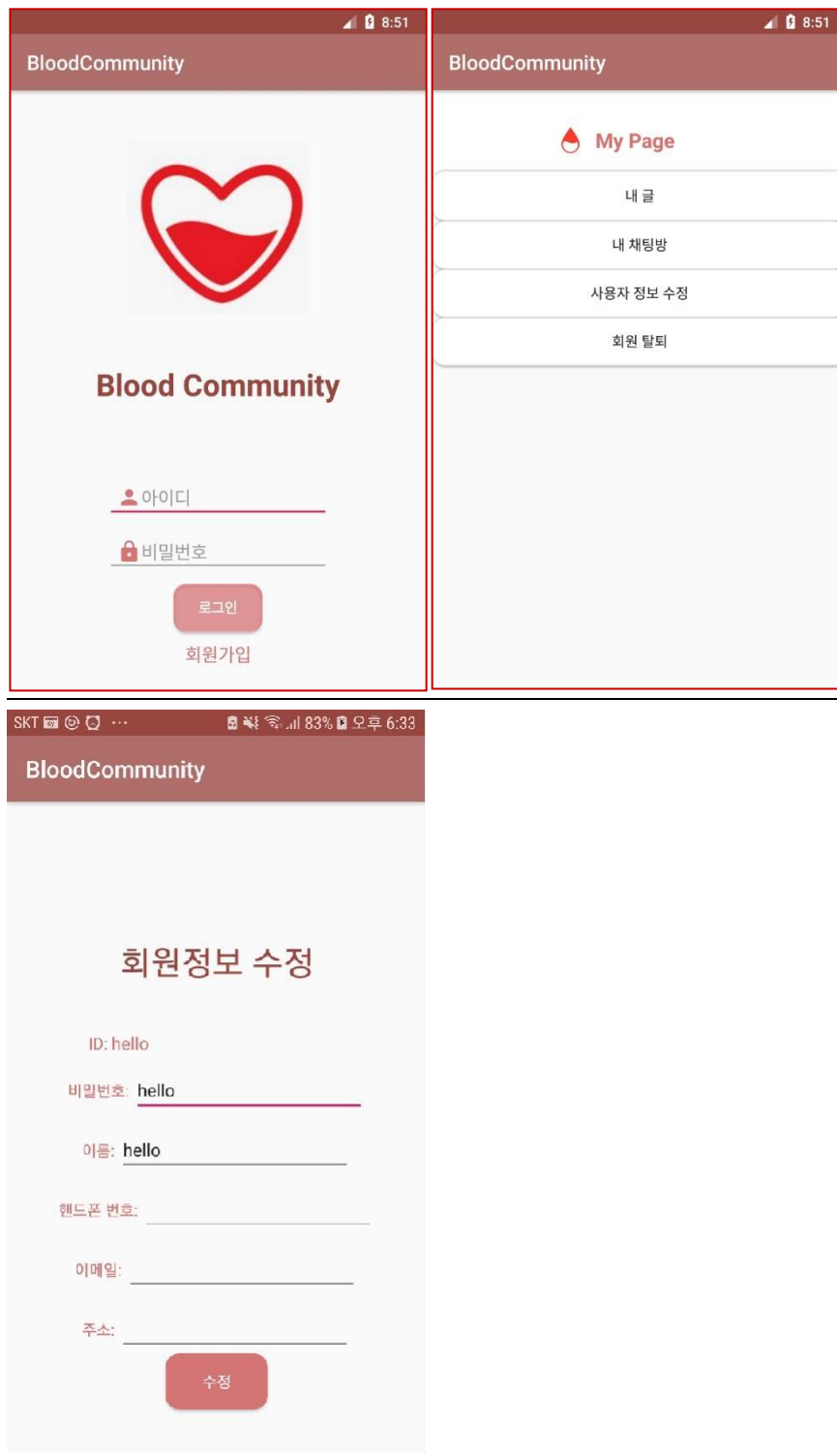
    @Override
    protected Void doInBackground(Void... voids) {
        try {
            Document doc = Jsoup.connect( url: "https://www.bloodinfo.net/event.do?action=detail&eventno=154213&current..."
            String title = doc.title();
            Element UriEle=doc.select( cssQuery: "img[alt=전국 헌혈의 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 URL 과 연결하고, 웹사이트의 html 코드를 이용하여 이미지 경로에 맞는 이미지를 가져온 후 실행하면 원하는 이미지가 크롤링됩니다.

실행결과 이미지





4. 방류시스템 데이터 보안을 위한 모바일 앱 개발(블록체인 적용)

프로젝트 개요

- 명칭: SuLock
- 개발 인원: 6 (개발자 3 명)
- 맡은 부분: 레이아웃, Blockchain SDK 적용
- 참여 기간: 2020.02~2020.02
- 개발 환경: 안드로이드 스튜디오(JAVA), SDK 이용
- 프로젝트 링크: <https://github.com/psyong99/SuLock>
- Certificate : https://drive.google.com/file/d/16_NbSU9_nHIUBnB1NIuH8-uOT7dmYkoc/view?usp=sharing

소개

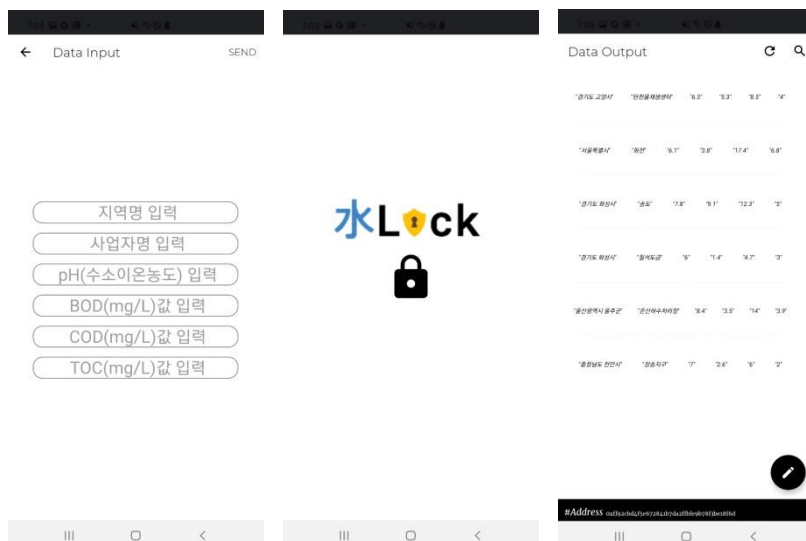
방류시스템의 데이터 위변조를 통해 무분별하게 오염수를 방류하는 문제를 해결하기 위해, 데이터 위변조가 불가능하도록 블록체인을 적용한 모바일 앱을 개발했습니다. 사용자가 데이터 수치를 입력하면, 블록체인에 데이터가 저장되고 저장된 데이터를 확인할 수 있는 서비스입니다.

프로젝트 목적

데이터의 위변조를 막고 안전하게 데이터가 저장됨을 확인하기 위해 개발했습니다. 블록체인은 데이터 보안이 가능하다는 점을 활용했습니다.

사용기술

- ✓ Java
- ✓ BlockChain SDK



+) Cloud 자격증

https://www.credly.com/badges/c25912b4-bba2-4b37-b169-095a1e60eff6/public_url

