

『기초프로젝트』 프로젝트 최종보고서

프로젝트 제목	수산물 위판정보 프로그램				작성년월일	2018/12/07
프로젝트 팀 이름	학번	2017243034	2017243126	2017243038	2017243044	2017243119
(해시태그 팀) 기여도 (김은선>송유빈>김이슬=박지현>허선)	성명	김은선	송유빈	김이슬	박지현	허선
	역할	로그인,회원가입,관리자,가격 검색.버튼 이벤트 연결 서버,디비 연결, 오픈API팀장	gui(메인,검색, 버튼 아이콘,디자인, 화면 이벤트 연결) ,중간ppt, 최종ppt제작, 최종보고서 작성	오픈API,물고기디비	gui(회원가입, 로그인, 화면 이벤트 연결)	gui(관리자)

1. 프로젝트 요약

여러 사이트에서 정보를 찾을 필요 없이 수산물의 정보를 이름, 제철, 서식지, 가격 순으로 제공하여 사용자가 원하는 정보를 알 수 있으며, (어류 기준) 한 마리 당 가격을 제공함으로 업계관련 종사자가 아니더라도 실시간으로 단가를 알 수 있게 하여 좀 더 적절한 가격으로 효율적인 소비를 할 수 있게 도와준다.

2. 프로젝트 내용

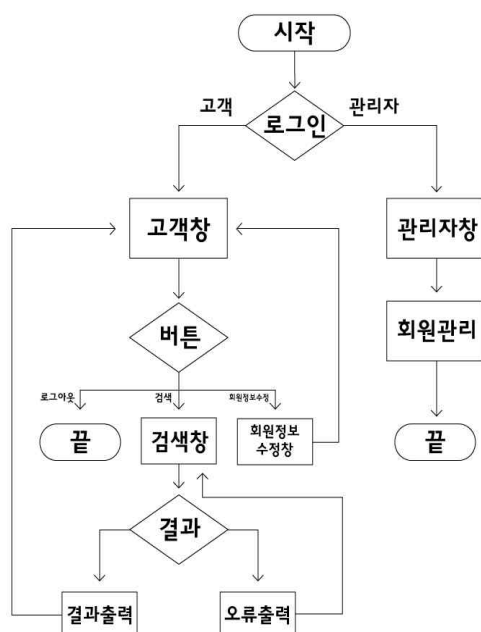
가. 프로젝트 제목

수산물 위판정보 프로그램

나. 프로젝트 목표

수산물의 정보(이름, 제철, 서식지 가격)을 제공하여 사용자가 원하는 정보를 쉽게 찾을 수 있게 하는 것

다. 프로젝트 사용된 알고리즘 및 순서도



라. 프로젝트에 사용된 주요 기술 현황

<Database>

Connection conn = null; //데이터베이스와 연결할 때 쓰일 변수

PreparedStatement stmt = null;

ResultSet rs=null; //결과를 가져온다.

String sql;//sql문을 적는다.

conn을 통해 데이터베이스와 연결한 후

stmt=conn.prepareStatement(sql);는 query를 실행할 준비를 한다.

rs=stmt.executeQuery();는 결과를 가져온다.

rs.next를 통해 데이터를 하나씩 읽어오고 ArrayList에 rs.getString("컬럼이름")을 통해 데이터를 하나씩 넣는다.

<webconnecton>

- 오픈 api의 주소값을 urlstr에 string 타입으로 저장하고, 그 값을 이용해 url 객체를 만든다.
httpconnection으로 url을 연결한다.

<webtest>

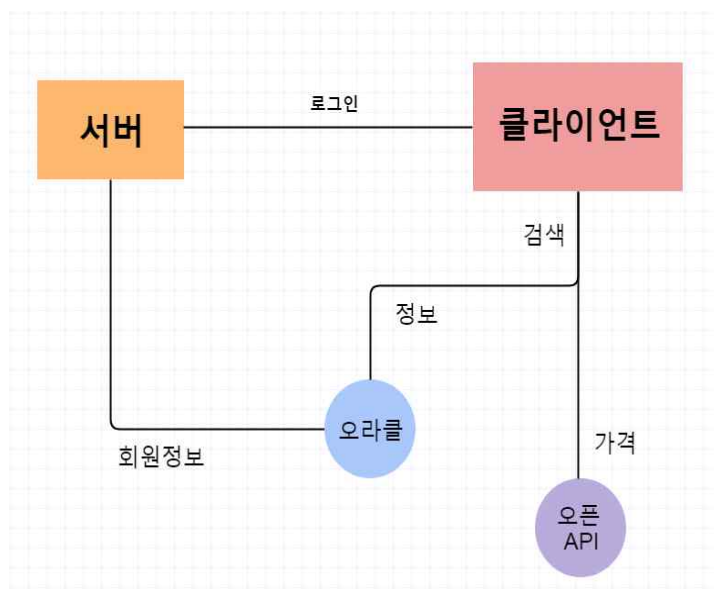
- json parser를 이용해 파싱을 하는데, json 문서의 가장 상위에 있는 response를 object에 넣고, 필요한 내용이 들어있는 item을 파싱하기 위해 item의 상위인 body부분을 object에 넣는다.

이후 item을 배열에 넣고, 필요한 값을 가져온다.

<Gui>

- dispose(); // 창에서 창으로 넘어갈 때 이전 창을 닫게 해주는 함수
setResizable(false); // 창 크기를 고정하는 함수

<시스템 구성도>



3. 프로젝트 수행결과물

가. 프로젝트 전체 개별 결과물 <* 결과물, 실행모습 등>

허선 (GUI창 만들기- 검색창 초안 어류, 관리자 환경설정 창, 관리자 회원관리 초안 창)



GUI초안이라 초반에 있었으나 프로젝트 구체화 중 수정되어 현재 존재하지 않음.

김이슬 (오픈 api 연결, 물고기 DB 제작)

오픈api 연결

```
package test;

import java.io.BufferedReader;

public class WebConnection{
    String result;
    Scanner sc = new Scanner(System.in);
    WebConnection() throws Exception{
        Calendar cal =Calendar.getInstance();
        BufferedReader br=null;
        int year = cal.get(cal.YEAR);
        int month = cal.get(cal.MONTH)+1;
        int date = cal.get(cal.DATE)-1;
        String dates="";
        if(date<10)
        {
            dates = "0"+String.valueOf(date);
        }
        else
        {
            dates = String.valueOf(date);
        }
        String years = String.valueOf(year);
        String months = String.valueOf(month);
        String fishname = null;
        System.out.println("물고기 이름 : ");
        fishname = sc.nextLine();

        String encodeResult = URLEncoder.encode(fishname, "UTF-8");
        System.out.println(year);
        System.out.println(month);
        System.out.println(date);
        System.out.println((year)+(months)+(date));

        try {
            String urlstr="http://apis.data.go.kr/1192000/openapi/service/ManageAcst0120Service/getAcst0120List?Serv
            URL url = new URL(urlstr);
            HttpURLConnection urlconnection=(HttpURLConnection)url.openConnection();
            urlconnection.setRequestMethod("GET");
            br=new BufferedReader(new InputStreamReader(urlconnection.getInputStream(),"utf-8"));
            String line;
            while(((line)=br.readLine())!=null)
            {
                result=line+"\n";
            }
        }catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

```

package test;
import java.io.*;

public class webtest {

    public static void main(String[] args) throws Exception {

        WebConnection wc = new WebConnection();
        String json=wc.result;
        System.out.println(json);
        if(json!=null) {
            System.out.println(json);
            JSONParser parser = new JSONParser();
            JSONObject obj=(JSONObject)parser.parse(json);
            JSONObject response=(JSONObject)obj.get("response");
            JSONObject body=(JSONObject)response.get("body");
            JSONArray item=(JSONArray)body.get("item");
            for(int i=0;i<item.size();i++)
            {
                JSONObject tmp = (JSONObject)item.get(i);
                String name=((String)tmp.get("mprcStdCodeNm"));
                String price=((String)tmp.get("csmtAmount"));
                String place=((String)tmp.get("csmtmktNm"));
                System.out.println("*****"+(i+1)+"번째 검색 결과*****");
                System.out.println("이름 : "+name);
                System.out.println("가격: "+price+"원");
                System.out.println("유통장: "+place);
            }
        }
        else
        {
            System.out.println("2");
        }
    }
}

```

```

1 select *
2 from fish

```

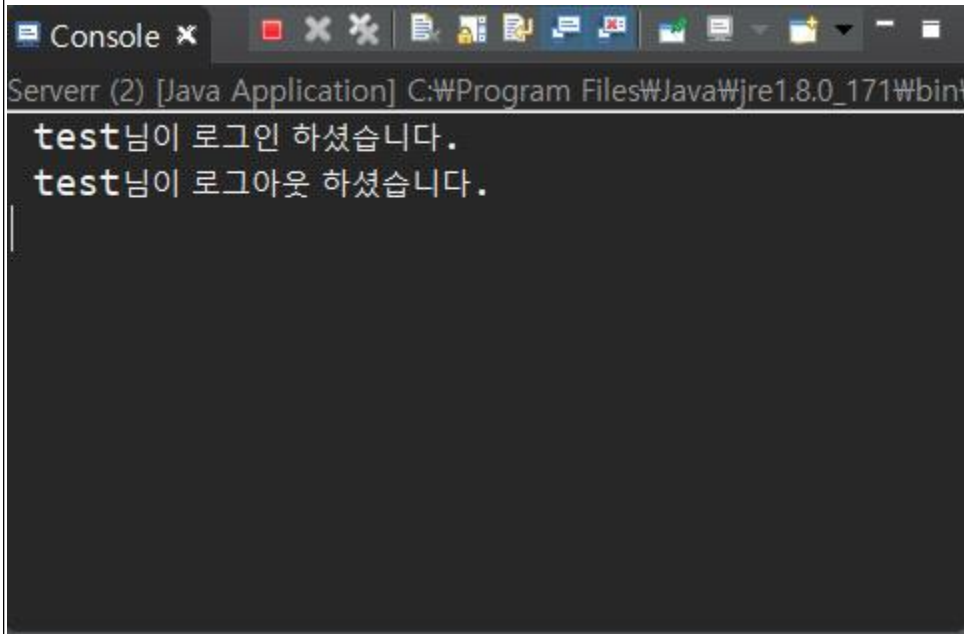
스크립트 출력 x | 실행 결과 x

SQL | 50개의 행이 인출됨(0.019초)

NAME	LENGTH	REGION	SEASON
1 입술무늬갑오징어	38	남서태평양, 남동중국해, 일본	3~5월
2 홍가자미	26~30	한국, 일본북부, 북서태평양	1월~5월
3 다금바리	90	제주도	11월~2월
4 참홍어	100	북서태평양	3월~6월
5 통치	40	남해, 서태평양, 인도양	6월~8월
6 가리맛조개 (맛)	10	남해, 서해 갯벌	7월~8월
7 군평선이	27	서해, 남해, 동중국해, 대만	3월~10월
8 덕대	60	동중국해, 대만, 한국, 일본	6월~9월
9 매통이	50	북서태평양, 인도양, 아프리카 연해	4월~5월
10 홍머	150	한국, 남일본, 동중국해	11월~4월
11 달강어	30	한국, 일본, 남중국해	3월~6월
12 등가시치	47	북서태평양 온대해역	5월~7월
13 양태류	60	남해, 대만, 호주, 인도양	6월~8월
14 눈볼대	30	남해, 인도양, 서태평양	12월~2월
15 꼬치고기류	30	한국, 일본남부, 동중국해, 인도양	11월~1월
16 다랑어류	300	동중국해, 동해, 태평양	4월~6월
17 붉은메기	70	한국, 일본 등 서태평양	4월
18 은대구	80	한국, 일본, 미국 등	10~12월
19 피뿌고등	15	한국, 일본, 중국 북부	3월~4월
20 꼼치	45	한국 동남부, 일본	11월~2월
21 닭새우	25	한국, 일본, 중국 등 서태평양	9월~12월
22 참가자미	40	동해, 남해, 동중국해	3월
23 숭어	60	한국, 중국, 러시아, 일본	11월~4월
24 중하	8~13	한국, 중국, 일본	9월~11월
25 눈강달미	15	서해, 동중국해, 일본 남부	5월~6월
26 기타낙지류	70	한국, 일본, 중국	9월~2월
27 가시발새우	12	한국, 중국, 일본, 대만, 필리핀 등	9월~11월
28 옥돔	45	남해, 일본 남부 등 서태평양 열대지역	11월
29 박대	50	서해, 동중국해 등 아열대 지역	10월~12월
30 다시마	150~350	한국, 일본, 캄차카반도 등 태평양 연안	7월~9월
31 용가자미	45	동중국해, 일본, 한국	2월~5월
32 적정미	17	한국, 중국, 일본	11월~2월
33 물가자미	40	한국, 일본 등 북서태평양 아열대 지역	4월~5월

김은선 (서버 연결, Gui와 DB 연결[회원관리, 로그인, 회원관리], 오픈 api와 검색창 연결)
서버

클라이언트에서 로그인과 로그아웃을 하면 서버에서 알려준다.



```
Server (2) [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\
test님이 로그인 하셨습니다.
test님이 로그아웃 하셨습니다.
```

관리자 창의 JTable

데이터 베이스에서 가져온 데이터들을 크기가 14인 String 배열에 넣은 후 JTable로 출력 한다.
회원가입은 최대 14명까지만 가능하다.



이름	ID	주소	이메일	핸드폰
김은선	ghd	전역	zxc@naver.com	01045674567
김은선	test	서울시	asd@naver.com	01012341234

회원가입

중복확인을 하지 않으면 버튼에 중복확인 해주세요 라는 메시지와 함께 회원가입이 되지 않는다.
만약 아이디가 중복된 아이디라면 중복된 아이디라고 알려준다.

회원가입

* 이름 이메일

* ID 주소

ID 중복확인용 확인해주세요!

* PW * 휴대폰

* 는 필수 입력사항입니다.

←

회원가입

* 이름 이메일

* ID 주소

중복된 아이디 입니다.

* PW * 휴대폰

* 는 필수 입력사항입니다.

←

중복확인을 하면 버튼의 메시지가 확인으로 바뀌고 확인을 누르면 정보들이 데이터베이스로 들어가고 회원가입이 완료된다.

회원가입

* 이름 이메일

* ID 주소

사용가능한 아이디 입니다.

* PW * 휴대폰

* 는 필수 입력사항입니다.

←

워크시트: 질의 작성기

```
SELECT * FROM customer
commit
```

스크립트 출력 x | 질의 결과 x

SOL | 인출된 모든 행: 3(0.003초)

	CUSTID	CUSTPA	EMAIL	PHONE	NAME	ADDRESS
1	ghd	1234	zxc@naver.com	01045674567	김길동	천안
2	jingu	jingu	jingu@gmail.com	01012351235	노진구	부산
3	test	test	asd@naver.com	01012341234	김한국	서울시

임시비밀번호 발급

이름, 아이디, 핸드폰번호를 입력한 후 serch를 누르면 100000~999999까지의 랜덤숫자를 임시 비밀번호로 발급해 준다. 데이터베이스에 있는 비밀번호도 임시로 발급된 비밀번호로 바뀐다.

← PW찾기

이름

이한국

ID

test

휴대폰

01043214321

search

귀하의 임시PW 는

407403

입니다.

CUSTID	CUSTPA	EMAIL	PHONE	NAME	ADDRESS
1	test	563815	qwe@naver.com	01043214321	이한국 인천

회원정보변경

변경할 회원정보를 입력한 후 확인을 누르면 데이터베이스에서 데이터가 변경됩니다.

회원관리

회원정보 수정

이름

이한국

이메일

qwe@naver.com

ID

test

주소

인천

PW

.....

휴대폰

01043214321

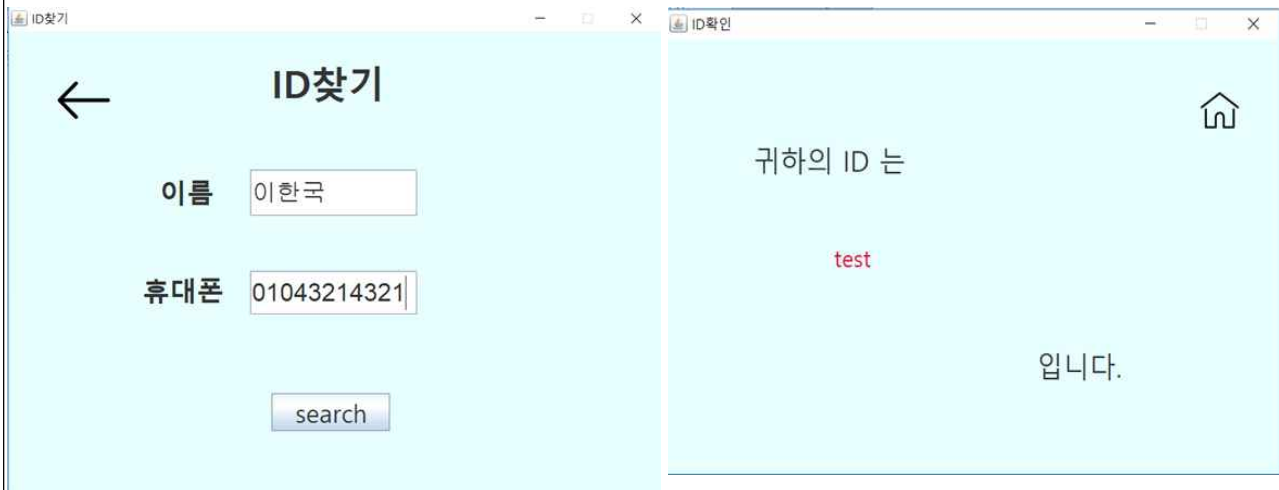
확인

🏠

변경전						변경후																													
<table><tr><th>CUSTID</th><th>CUSTPA</th><th>EMAIL</th><th>PHONE</th><th>NAME</th><th>ADDRESS</th></tr><tr><td>1</td><td>test</td><td>asd@naver.com</td><td>01012341234</td><td>김한국</td><td>서울시</td></tr></table>						CUSTID	CUSTPA	EMAIL	PHONE	NAME	ADDRESS	1	test	asd@naver.com	01012341234	김한국	서울시	<table><tr><th>CUSTID</th><th>CUSTPA</th><th>EMAIL</th><th>PHONE</th><th>NAME</th><th>ADDRESS</th></tr><tr><td>1</td><td>teest</td><td>qwe@naver.com</td><td>01043214321</td><td>이한국</td><td>인천</td></tr></table>						CUSTID	CUSTPA	EMAIL	PHONE	NAME	ADDRESS	1	teest	qwe@naver.com	01043214321	이한국	인천
CUSTID	CUSTPA	EMAIL	PHONE	NAME	ADDRESS																														
1	test	asd@naver.com	01012341234	김한국	서울시																														
CUSTID	CUSTPA	EMAIL	PHONE	NAME	ADDRESS																														
1	teest	qwe@naver.com	01043214321	이한국	인천																														

아이디찾기

이름과 핸드폰번호를 입력한 후 search 버튼을 누르면 아이디를 찾아줍니다.



The image shows two side-by-side browser windows. The left window, titled 'ID찾기', has a light blue background. It features a back arrow icon, a title 'ID찾기', and two input fields: '이름' (Name) with the value '이한국' and '휴대폰' (Mobile Phone) with the value '01043214321'. Below these is a 'search' button. The right window, titled 'ID확인', also has a light blue background. It displays '귀하의 ID 는' (Your ID is) in black, 'test' in red, and '입니다.' (is) in black. A home icon is in the top right corner.

로그인

아이디와 비밀번호를 입력하고 로그인을 누르면 아이디가 관리자일 경우 관리자 창을 보여주고 아이디가 고객일 경우 고객창을 보여준다.

고객일 경우



The image shows two side-by-side browser windows. The left window, titled '로그인' (Login), has a light blue background. It features input fields for 'ID' (value: 'jingu') and 'PW' (value: '.....'). Below these are buttons for '회원가입' (Sign Up), 'ID 찾기' (Find ID), and 'PW 찾기' (Find PW). At the bottom is a 'Login' button and a home icon. The right window, titled '수산물 위판정보 프로그램' (Seafood Auction Information Program), has a light blue background. It features a large blue fish illustration. On the left is a menu with buttons: '어류' (Fish), '해조류' (Seaweed), '어패류' (Seafood), '갑각류' (Crustaceans), and '연체동물' (Mollusks). On the right is a user profile for 'jingu님' with a 'LOGOUT' button. At the bottom is a footer with small text: '©2014년 충남수산시장협동조합. 2014년 1월 7일. 충청남도 천안시 동남구 410호'.

관리자일 경우



The image shows two side-by-side browser windows. The left window, titled '로그인' (Login), has a light blue background. It features input fields for 'ID' (value: 'Administer') and 'PW' (value: '...'). Below these are buttons for '회원가입' (Sign Up), 'ID 찾기' (Find ID), and 'PW 찾기' (Find PW). At the bottom is a 'Login' button and a home icon. The right window, titled '수산물 위판정보 프로그램' (Seafood Auction Information Program), has a light blue background. It features a large blue fish illustration. On the left is a menu with buttons: '어류' (Fish), '해조류' (Seaweed), '어패류' (Seafood), '갑각류' (Crustaceans), and '연체동물' (Mollusks). On the right is a user profile for 'Administer' with a 'LOGOUT' button. At the bottom is a footer with small text: '©2014년 충남수산시장협동조합. 2014년 1월 7일. 충청남도 천안시 동남구 410호'.


가격 검색

가격을 알고 싶은 물고기를 검색하면 가격과 제철 서식지를 알려준다. 만약 검색결과가 없다면 검색결과가 없다는 창을 띄워준다.



박지현 (GUI 창 만들기- 로그인,회원가입,아이디찾기,비밀번호찾기)

로그인 창 (ID • PW 텍스트필드 추가 , 회원가입 • ID찾기 • PW찾기 • Login 버튼생성)



회원가입 창 (이름 • ID • PW • 이메일 • 주소 • 휴대폰 텍스트필드 추가, 확인버튼생성)
로그인 창에서 회원가입버튼 클릭 시 나오는 창

회원가입 창 (회원가입 폼)의 스크린샷입니다. 창 제목은 '회원가입'입니다. 폼에는 다음과 같은 요소가 포함되어 있습니다:

- * 이름**: 텍스트 입력 필드
- 이메일**: 텍스트 입력 필드
- * ID**: 텍스트 입력 필드, 옆에 **중복확인** 버튼이 있습니다.
- 주소**: 텍스트 입력 필드
- * PW**: 텍스트 입력 필드
- * 휴대폰**: 텍스트 입력 필드
- ID 중복확인을 확인해주세요!
- * 는 필수 입력사항입니다.
- ← (뒤로가기 버튼)
- 확인** (확인 버튼)

(회원가입- 중복확인 하지 않았을 경우에 뜨는 창)

회원가입 창 (중복확인 메시지)의 스크린샷입니다. 창 제목은 '회원가입'입니다. 폼에는 다음과 같은 요소가 포함되어 있습니다:

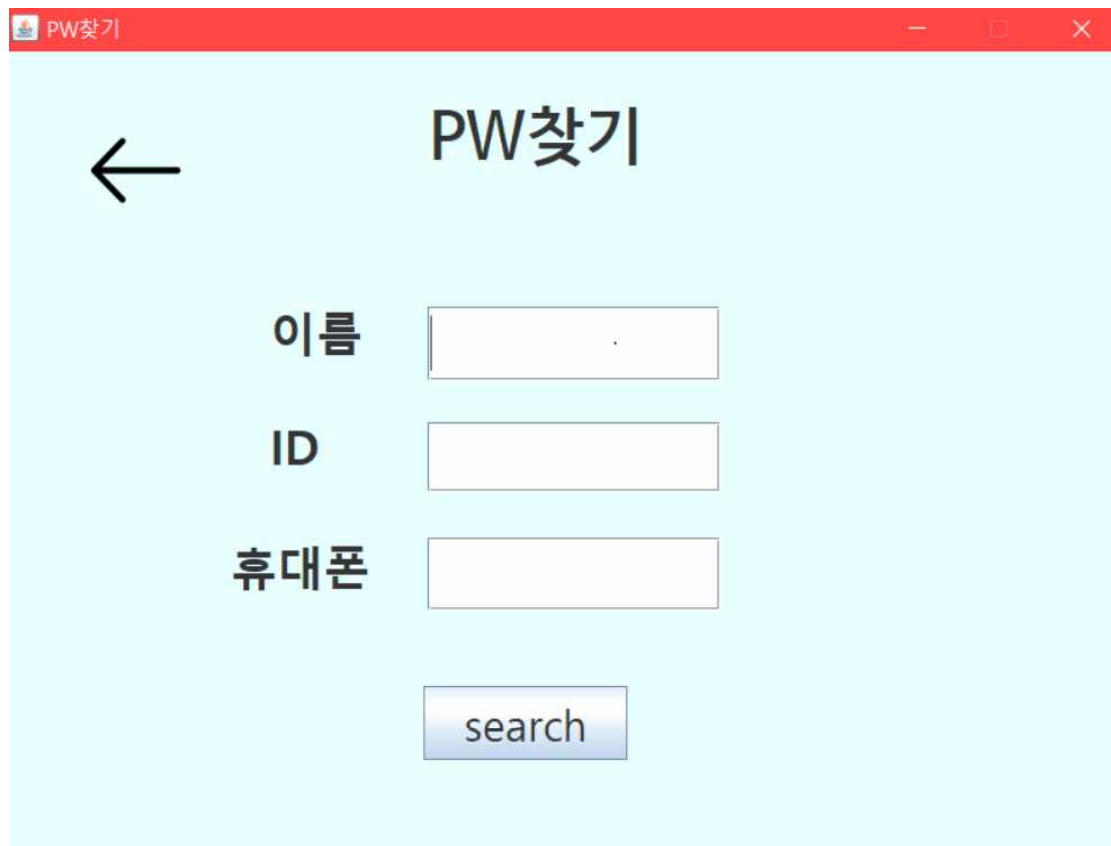
- * 이름**: 텍스트 입력 필드
- 이메일**: 텍스트 입력 필드
- * ID**: 텍스트 입력 필드, 옆에 **중복확인** 버튼이 있습니다.
- 주소**: 텍스트 입력 필드
- * PW**: 텍스트 입력 필드
- * 휴대폰**: 텍스트 입력 필드
- ID 중복확인을 확인해주세요!
- * 는 필수 입력사항입니다.
- ← (뒤로가기 버튼)
- 중복확인 해주세요** (확인 버튼)

ID 찾기 창 (이름 • 휴대폰 텍스트 필드 , search 버튼생성)
로그인창에서 ID찾기 클릭시 나오는 창



The image shows a web browser window titled "ID찾기". The page has a light blue background. At the top left, there is a black left-pointing arrow. To its right, the title "ID찾기" is displayed in large black font. Below the title, there are two text input fields. The first field is preceded by the label "이름" (Name) in bold black font. The second field is preceded by the label "휴대폰" (Mobile Phone) in bold black font. Below these fields, there is a blue button with the word "search" in white text.

PW 찾기 창 (이름 • ID • 휴대폰 텍스트 필드 , search 버튼생성)
로그인창에서 PW찾기 클릭시 나오는 창



The image shows a web browser window titled "PW찾기". The page has a light blue background. At the top left, there is a black left-pointing arrow. To its right, the title "PW찾기" is displayed in large black font. Below the title, there are three text input fields. The first field is preceded by the label "이름" (Name) in bold black font. The second field is preceded by the label "ID" in bold black font. The third field is preceded by the label "휴대폰" (Mobile Phone) in bold black font. Below these fields, there is a blue button with the word "search" in white text.

송유빈 (GUI창 만들기 - 기본, 회원, 관리자 메인창 3개, 검색창 5개, 회원정보 수정 창, ID찾기 결과 창, PW 찾기 결과 창, 경고창, 관리자 설정 창, 모든 창 이벤트 연결, 모든 창 디자인 제작)

기본 메인 창 (레이블에 이미지 삽입, 텍스트필드로 학교 주소, 프로그램 이름, 로그인 버튼, 버튼 이벤트 연결)



회원 메인 창 (레이블에 이미지 삽입, 레이블로 회원 아이디 삽입, 텍스트필드로 학교 주소, 프로그램 이름, 어류~연체동물 버튼, 로그아웃 버튼, 회원정보 수정 버튼, 버튼 이벤트 연결)

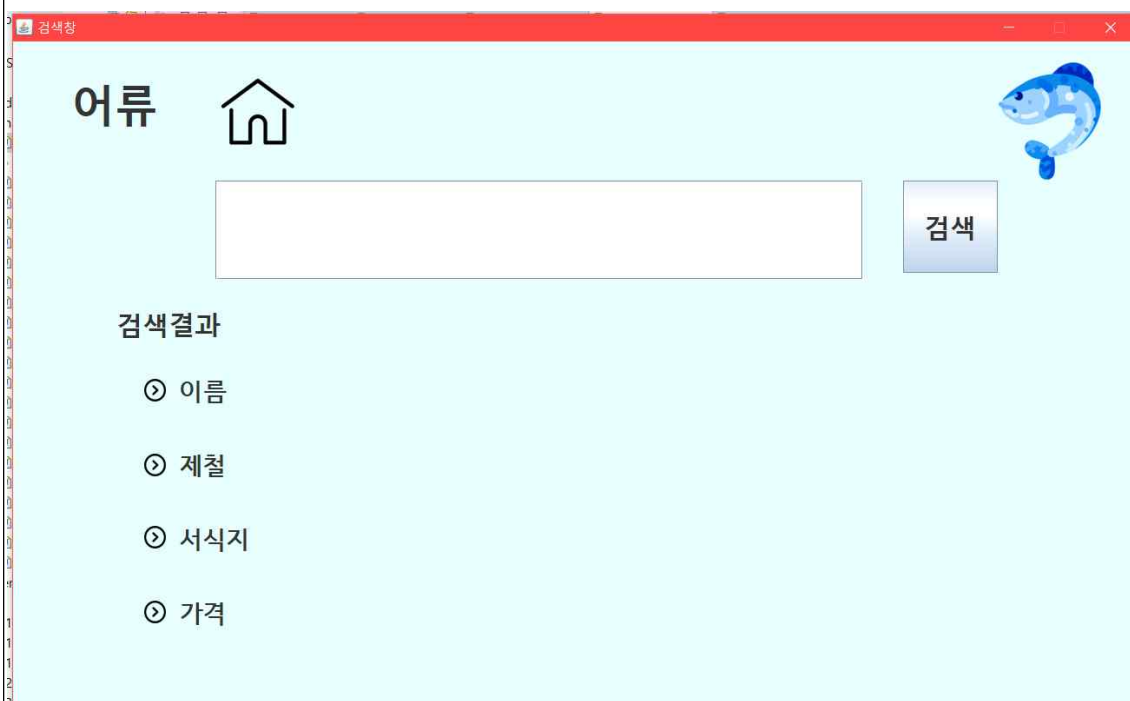


관리자 메인 창 (레이블에 이미지 삽입, 레이블로 Administer 삽입, 텍스트필드로 학교 주소, 프로그램 이름, 어류~연체동물 버튼, 로그아웃 버튼, 관리자 설정 버튼, 버튼 이벤트 연결)



검색창 (레이블 삽입, 홈 버튼 삽입, 검색 텍스트 필드, 검색 버튼 삽입, 이미지 삽입, 버튼 이벤트 연결)


어류 창



해조류 창, 어패류 창

검색창

해조류 




검색

검색결과

- ☐ 이름
- ☐ 재철
- ☐ 서식지
- ☐ 가격

검색창

어패류 




검색


검색결과

- ☐ 이름
- ☐ 재철
- ☐ 서식지
- ☐ 가격

갑각류 창, 연체동물 창

검색창

갑각류 





검색

검색결과

- ☐ 이름
- ☐ 재철
- ☐ 서식지
- ☐ 가격

검색창

연체동물 



검색

검색결과

- ☐ 이름
- ☐ 재철
- ☐ 서식지
- ☐ 가격

회원정보 수정 창 (각 항목의 텍스트필드 삽입, ID 항목 레이블 삽입, 버튼 삽입, 버튼 이벤트 연결)

회원관리

회원정보 수정

이름


이메일

ID yobin4413

주소

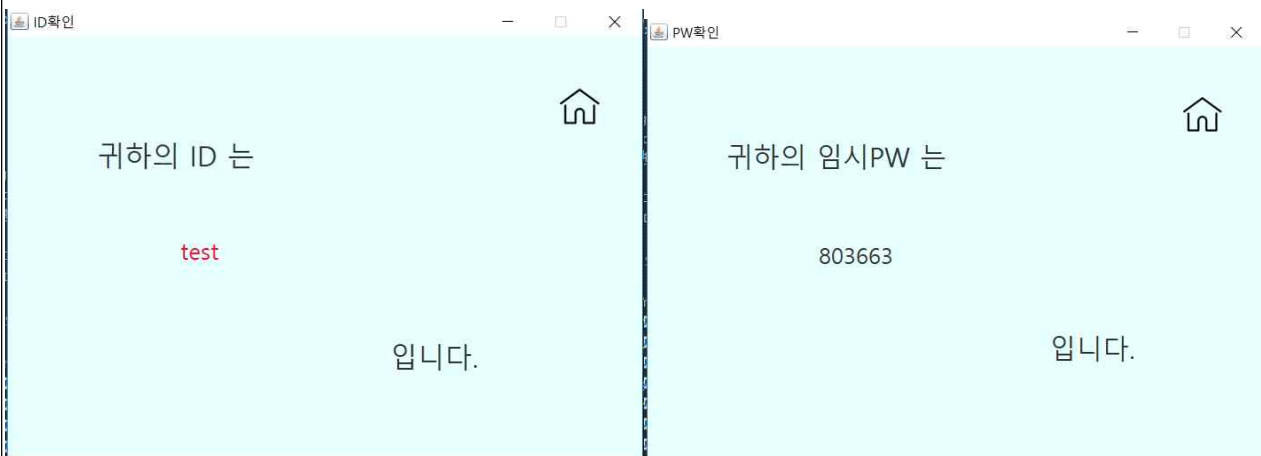
PW

휴대폰



확인

ID 찾기 결과 창, PW 찾기 결과 창 (레이블 삽입, 홈 버튼 삽입, 버튼 이벤트 연결)



경고 창 (이미지 삽입, 레이블 삽입, 버튼 삽입, 버튼 이벤트 연결)



관리자 설정 창 (버튼 삽입, 레이블 삽입, 버튼 이벤트 연결)



나. 개발 결과물의 주요 부분 기술개발 내용

개발 시스템

Eclipse 로는 자바를 사용해 Gui에 필요한 스윙과 서버, 스레드를 사용하였다.

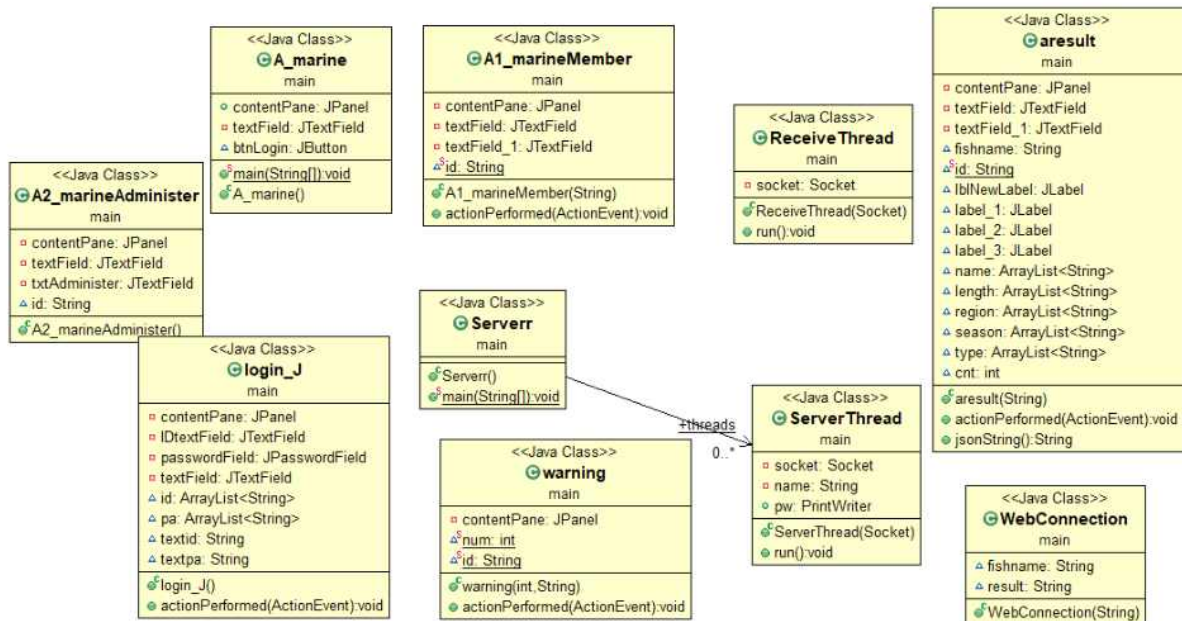
Oracle로는 데이터베이스로 고객정보 DB와 물고기정보 DB를 사용하였다.

JSON은 오픈 Api에서 제공하는 정보가 json형태로 출력되기 때문에 사용하였다.

주요부분 기술개발

찾으려는 수산물의 위판정보를 오픈 Api를 통해 실행일 전날의 데이터를 확인 할 수 있고, 하루 단위로 정보가 업데이트되기 때문에 검색할 시 매일 위판정보가 달라진다. 따라서 api를 받아와 json의 형태로 출력 받아 프로그램을 실행한다.

<클래스 다이어그램>




4. 프로젝트 수행결과

- 프로젝트의 핵심 내용이 오픈 api로 받아온 위판정보를 프로그램으로 보여주는 것이 제일 중요한 핵심이기 때문에 검색 결과로 나오는 모습


어류 결과 창



해조류 결과 창 - 데이터가 없어서 가격이 표시되지 않음



해조류



검색

검색결과

- 이름: 다시마
- 제철: 7월~9월
- 서식지: 한국, 일본, 캄차카반도 등 태평양 연안
-

```
http://apis.data.go.kr/1192000/openapi/service/ManageAcst0120Service/getAcst0120List?ServiceKey=Aty4CxFSXK4Q2Q3iqEGjy2
{"response":{"header":{"title":"위판장별 위탁판매 현황","resultCode":"03","resultMsg":"조회된 데이터가 없습니다"},"body":{"item":[],"
```

어패류 결과 창



어패류



검색

검색결과

- 이름: 키조개
- 제철: 7월~8월
- 서식지: 한국, 일본
- 가격: 13000원

갑각류 결과 창



검색창

갑각류 

검색결과

- ① 이름: 꽃게
- ① 제철: 4월~6월, 8월~12월
- ① 서식지: 한국, 일본, 중국해역
- ① 가격: 90000원

연체동물 결과 창



검색창

연체동물 

검색결과

- ① 이름: 문어
- ① 제철: 11~4월
- ① 서식지: 아메리카 북서쪽 태평양, 캘리포니아 남쪽, 일본 남쪽
- ① 가격: 42000원

5. 문제점 검토 및 향후 개선방향

- localhost로 되어있어서 다른 컴퓨터에서는 사용할 수가 없기 때문에 향후 어느 곳 에서나 사용 할 수 있도록 하는 방법을 연구해봐야겠다.
- 처음 시작할 때 GUI를 따로 만들었기 때문에 GUI클래스 파일이 너무 많아졌다. 향후 클래스의 수를 줄여야겠다.
- 검색이나 로그인등 텍스트 필드에 작성할 시 엔터를 무조건 쳐야한다. 만약에 엔터를 치지 않을 경우에 결과 값이 창에 나타나지 않는다.
- 관리자 권한을 가질 수 있는 요소를 늘려 관리자의 역할을 원활히 수행 할 수 있게 한다.

6. 프로젝트 수행 후기

처음 주제를 선정할 때에는 비교적 순탄하게 정해졌다. 크게 자바 GUI 부분과 디비(서버)로 나누어 각자 맡을 부분을 정했다.

자바 gui를 처음 시작 할 때는 시스템 창을 구성하는데 창의 개수가 많아서 좀 버거웠다. 다들 자바에 익숙하지 않아서 여러 방향으로 참고를 하느라 시간이 많이 걸렸다. 3명에서 나누어서 하다 보니 디자인이나 창 크기 등 세세한 부분을 이후에 통일시키고 수정하는 데 시간이 좀 걸렸다. 다음에 프로젝트를 한다면 세세한 부분 등을 미리 다 정한 후에 시작해야겠다는 생각이 들었다.

처음에는 창과 창을 자연스럽게 버튼을 누르면 창이 없어지는 게 아니라 창이 계속 쌓이게 되었다. 함수를 따로 찾아서 창과 창을 자연스럽게 연결하였다. 이미지를 삽입을 했던 노트북에서는 이미지가 잘 나왔지만 다른 조원들 노트북에서는 이미지가 나오지 않았던 문제가 있었다. 이로 인해서 이미지 삽입 코드 전체를 수정하였다. 또한 디자인을 수정하게 되면 모든 코드를 다시 넘겨줘야하는 불편함이 있었다.

DB로 쿼리문을 보낼 때 SELECT문은 오류가 없이 잘되었었지만 UPDATE문을 사용하면 오라클 데이터베이스에서 정보가 바뀌지 않았다. 이 오류를 해결하기 위해 많은 시간이 걸렸지만 해결하고 난 후 성취감을 느꼈다. 이번 프로젝트를 통해 버튼 이벤트 연결과 데이터베이스 연결을 능숙하게 할 수 있게 되었다.

처음 json을 연결할 때, 무작정 블로그만 보고 따라하다 보니 오류가 정말 많이 났었는데 코드를 조금씩 이해해 오류를 해결할 때는 굉장히 뿌듯했다. 이번 프로젝트로 자바 프로그래밍 실력이 조금 더 늘게 된 것 같고, 다른 프로젝트를 할 때도 활용할 수 있을 것 같다. swing은 제대로 해보지 않아서 나중에 따로 연습을 해봐야 할 것 같다.

7. 참고문헌

-데이터베이스 연결

<http://best421.tistory.com/47>

- GUI

<https://blog.naver.com/tmdgns7053?Redirect=Log&logNo=221313327484>

- [파싱] 4. API키 발급 및 JSON을 이용한 웹 파싱 : 네이버 블로그

<http://blog.naver.com/PostView.nhn?blogId=occidere&logNo=220799351272>

- 공공데이터포털 OPEN API 사용하는 방법

<http://jeong-pro.tistory.com/143>

-[Oracle] DB에서 excel파일 데이터 Import, Export

<http://tora-it-kingdom.tistory.com/8>

- GUI 버튼 이벤트 연결

<http://fors.tistory.com/184>

- 자바 GUI 이미지 버튼 만들기

<https://m.blog.naver.com/wari7i7/220853504127>