



객 체 지 향 프 로 그 래 밍 Ⅱ 1 분 반 0 7 조

부동산 매물 정보제공 시스템

김장

준원

근재

김김

상주

환의



역할 분
담



프로젝트
개요

문제 기술
비전
기능목록



시스
템

소프트웨어 구조
시스템 인터페이
스
사용자 인터페이
스
통신 인터페이스



특징 구현

부동산 정보 검색

엑셀 파일 파싱

로그인

아이디/비밀번호 찾기



프로젝트 평가

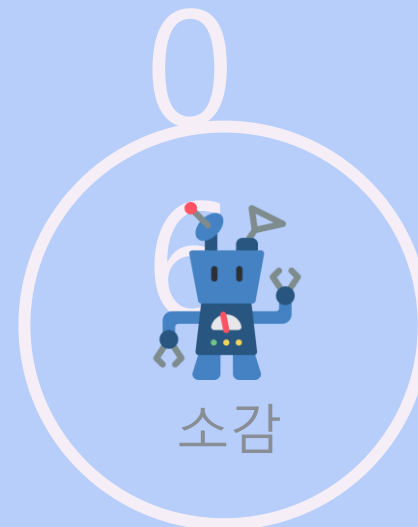
프로젝트 완성도

일정 계획 평가

역할 수행 평가

설계 구성요소

현실적 제한조건



소감

01역할 분담



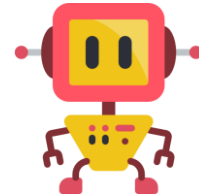
팀장 장윤재

아파트API파싱
오피스텔 API파싱
GUI 구현
테스팅
형상관리
코딩



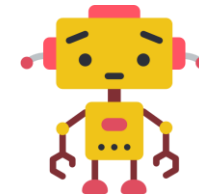
코딩 김준근

아이디어 제공
연립 다세대 API파싱
단독/다가구 API파싱
엑셀파일 파싱
코딩



디자인 김상환

자료조사
영상편집
GUI 설계
디자인 설계



서기 김주의

회의록 작성
GUI 설계
테스팅
중간/최종보고서 작성
PPT 제작

02 프로젝트 개요



● 문제 기술

직접 매물을 보러 다니기엔 시간과 힘이 많
이 듭
헛걸음하 시간 낭비



불편함을 줄여주고, 쉽고 간단하고 명확하게 매물
정보 제공!

● 비전

요즘 세대는 직접 찾으러 다니는 것보단 스마트폰이나 PC로



현대인에게 맞춰 불편함을 줄이고 요구사항을 반영하여
편리함 제공!

02 프로젝트 개요



기능 목록

- 01 공공 API를 이용하여 매물 정보 제공
- 02 원하는 매물 위치 검색
- 03 사용자 로그인/회원가입
- 04 아이디/비밀번호 찾기

client

Client
TcpClientTest
TcpEchoClient

login

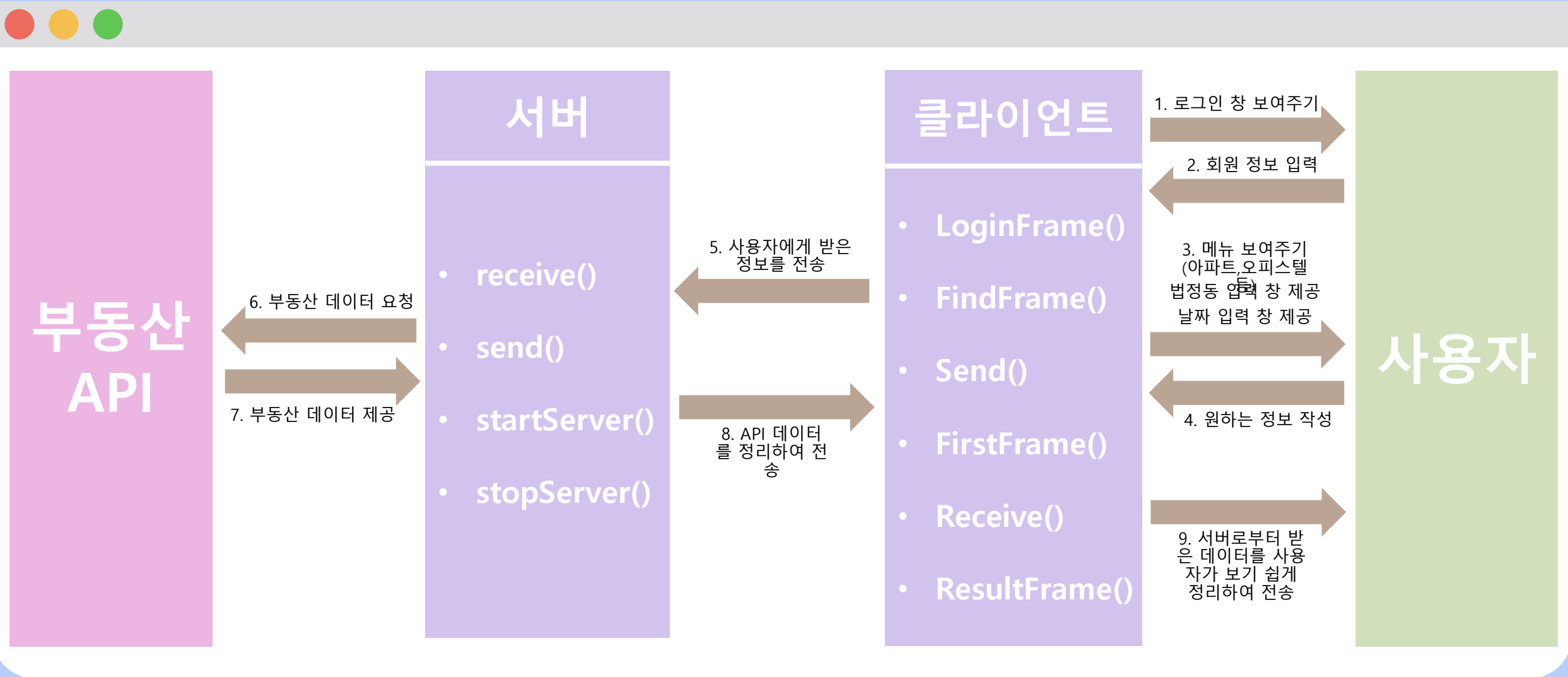
AptPrice
AptResultFrame
FindIDPasswordFrame
FirstFrame
LoginFrame
OftResultFrame
RegisterFrame
RhtRegisterFrame
ShtRegisterFrame

parsing

CategoryParsing
ExcelPs
ParsingApt
ParsingBase
ParsingOft
ParsingRht
ParsingSht
TestDrive

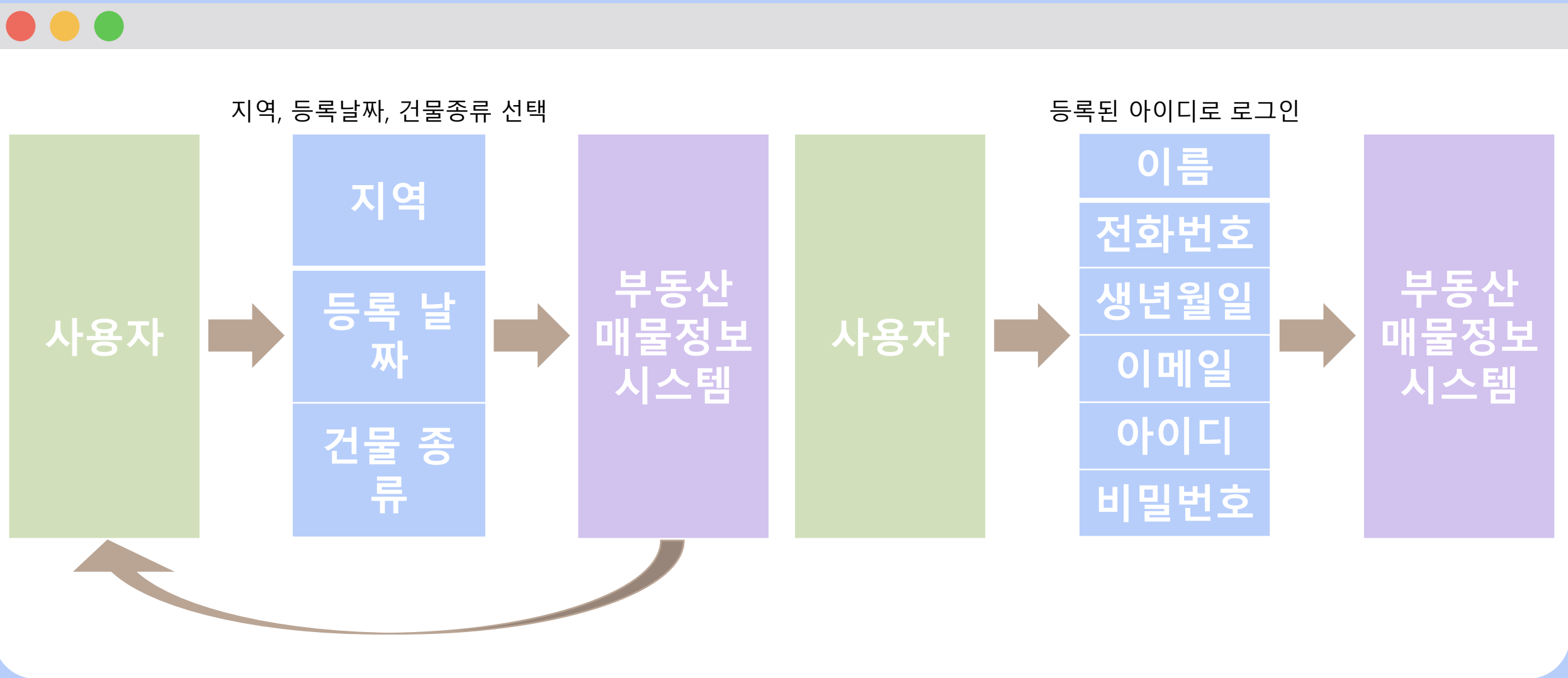
server

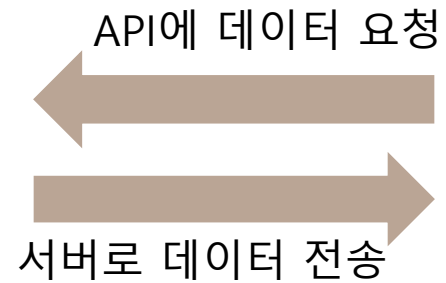
Server
TcpEchoServer
TcpServerTest



03 시스템

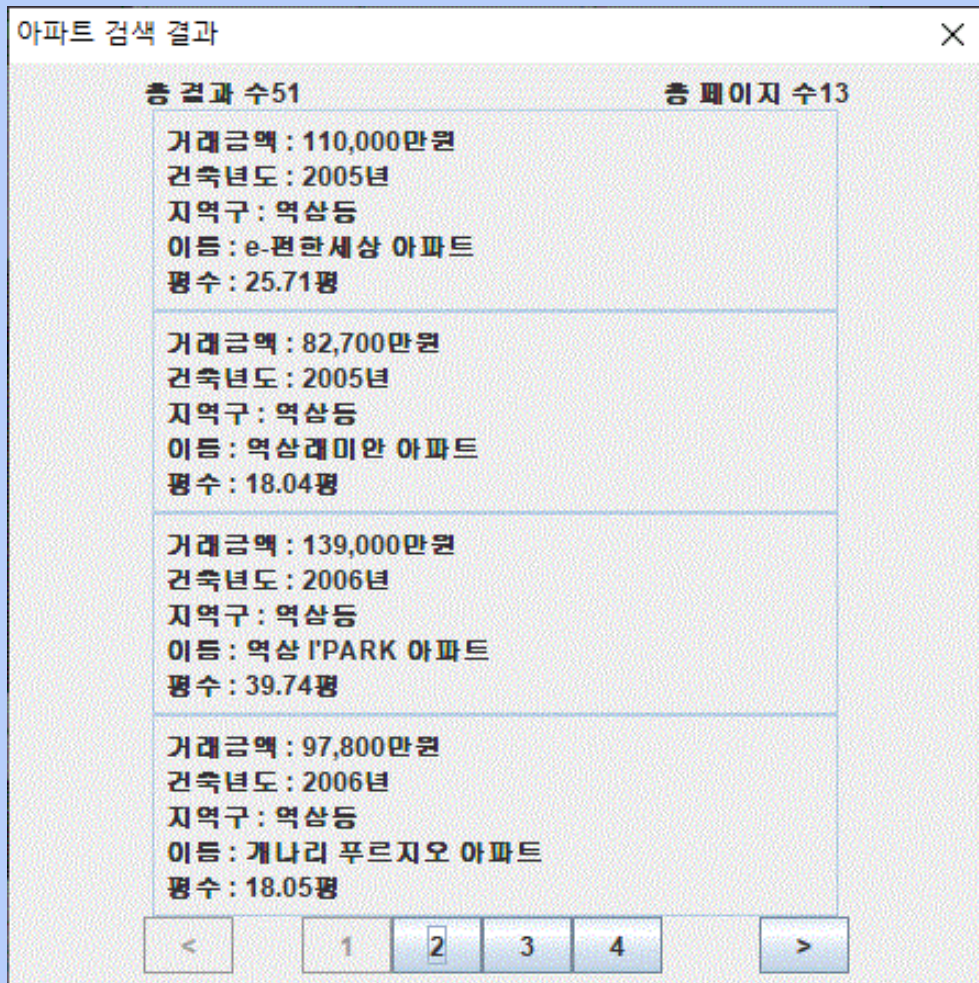
사용자 인터페이스





04특징 구현

부동산 정보 검색



기능

전국 매매정보 등을 검색하여 정보를 제공

요구사항

공공 API 파싱을 통해 아파트, 단독 다가구, 연립 다세대, 오피스텔의 부동산정보 요구

구현 방안

파싱 클래스 분할
GUI 창에서 주택 유형을 선택해 원하는 지역을 검색하면 GUI 창으로 부동산 정보가 나오도록 구현

URL을 만듦

```
public void setUrl() {  
    urlMaker.append(priceList); //시작 URL  
    urlMaker.append(cateGory); //아파트, 단독/다가구, 연립다세대의 종류를 선택  
    urlMaker.append("?LAWD_CD=");  
    urlMaker.append(reignCode); //int형으로 선언해준 지역 코드를 string형으로 형 변환  
    urlMaker.append("&DEAL_YMD=");  
    urlMaker.append(dealMonth);  
    urlMaker.append("&ServiceKey=");  
    urlMaker.append(serviceKey);  
    System.out.println(urlMaker);  
}
```

04특징 구현

부동산 정보 검색

● ● ● 파싱 한 아파트 매물 정보를 분류

```
public void contents(NodeList nodeList) {
    for (int i = 0; i < nodeList.getLength(); i++) {
        NodeList child = nodeList.item(i).getChildNodes();
        resultBuilding.setLength(0); //StringBuilder 를 초기화
        for (int j = 0; j < child.getLength(); j++) { //items 으로 둘러싸인 부분
            Node node = child.item(j);

            if (node.getNodeName().contains("거래금액")) { // item 으로 둘러싸인 부분
                // contains 와 equal 사용 고려
                String temp = node.getTextContent();
                temp = temp.trim(); //문자열의 앞 뒤 공백 제거 함수
                resultBuilding.append("거래금액 : ");
                resultBuilding.append(temp);
                resultBuilding.append("만원");
                resultBuilding.append("\n"); // 줄바꿈
            }
        }
    }
}
```

```
else if (node.getNodeName().contains("건축년도")) { //item 으로 둘러싸인 부분을 반복
    //contains 와 equal 사용 고려
    String temp = node.getTextContent();
    temp = temp.trim(); //문자열의 앞 뒤 공백 제거 함수
    resultBuilding.append("건축년도 : ");
    resultBuilding.append(temp);
    resultBuilding.append("년");
    resultBuilding.append("\n"); // 줄바꿈
}
else if (node.getNodeName().contains("법정동")) { //item 으로 둘러싸인 부분을 반복
    String temp = node.getTextContent();
    temp = temp.trim(); //문자열의 앞 뒤 공백 제거 함수
    if(temp.equals(area)) {
        chk=1;
    }
    resultBuilding.append("지역구 : ");
    resultBuilding.append(temp);
    resultBuilding.append("\n"); // 줄바꿈
}
else if (node.getNodeName().contains("아파트")) { //item 으로 둘러싸인 부분을 반복
    .
    .
    .
}
```

04특징 구현

엑셀 파일 파싱

4831033024	경상남도	거제시	남부면	다대리	19950101	[null]
4831033025	경상남도	거제시	남부면	갈곶리	19950101	[null]
4831034000	경상남도	거제시	거제면	[null]	19950101	[null]
4831034021	경상남도	거제시	거제면	법동리	19950101	[null]
4831034023	경상남도	거제시	거제면	내간리	19950101	[null]
4831034024	경상남도	거제시	거제면	외간리	19950101	[null]
4831034025	경상남도	거제시	거제면	옥산리	19950101	[null]
4831034026	경상남도	거제시	거제면	서정리	19950101	[null]
4831034027	경상남도	거제시	거제면	서상리	19950101	[null]
4831034028	경상남도	거제시	거제면	남동리	19950101	[null]
4831034029	경상남도	거제시	거제면	통상리	19950101	[null]
4831034030	경상남도	거제시	거제면	오수리	19950101	[null]
4831034031	경상남도	거제시	거제면	명진리	19950101	[null]
4831034032	경상남도	거제시	거제면	소량리	19950101	[null]
4831035000	경상남도	거제시	문덕면	[null]	19950101	[null]
4831035021	경상남도	거제시	문덕면	상동리	19950101	[null]
4831035022	경상남도	거제시	문덕면	시목리	19950101	[null]
4831035023	경상남도	거제시	문덕면	거림리	19950101	[null]
4831035024	경상남도	거제시	문덕면	산방리	19950101	[null]
4831035025	경상남도	거제시	문덕면	방하리	19950101	[null]
4831035026	경상남도	거제시	문덕면	하둔리	19950101	[null]
4831035027	경상남도	거제시	문덕면	어구리	19950101	[null]
4831035028	경상남도	거제시	문덕면	습역리	19950101	[null]
4831035029	경상남도	거제시	문덕면	학산리	19950101	[null]
4831036000	경상남도	거제시	사동면	[null]	19950101	[null]
4831036021	경상남도	거제시	사동면	사곡리	19950101	[null]
4831036022	경상남도	거제시	사동면	사동리	19950101	[null]
4831036023	경상남도	거제시	사동면	성포리	19950101	[null]
4831036024	경상남도	거제시	사동면	지석리	19950101	[null]
4831036025	경상남도	거제시	사동면	청곡리	19950101	[null]
4831036026	경상남도	거제시	사동면	오량리	19950101	[null]
4831036027	경상남도	거제시	사동면	덕호리	19950101	[null]
4831036028	경상남도	거제시	사동면	창호리	19950101	[null]
4831037000	경상남도	거제시	연초면	[null]	19950101	[null]
4831037021	경상남도	거제시	연초면	한내리	19950101	[null]
4831037022	경상남도	거제시	연초면	오비리	19950101	[null]
4831037023	경상남도	거제시	연초면	연사리	19950101	[null]
4831037024	경상남도	거제시	연초면	죽토리	19950101	[null]
4831037025	경상남도	거제시	연초면	다공리	19950101	[null]

기능

전국기준 38,000개의 지역 코드가 들어있는 엑셀 파일을 파싱 하여 지역별 코드 정보를 받아옴

요구사항

공공 API 파싱을 위해 지역코드가 필요한데 지역 코드 수가 너무 많아 엑셀 파일에서 읽어와야 함

구현 방안

엑셀 파일을 읽어 오기 위해 엑셀 파일을 다루는데 좋은 POI 라이브러리 사용

04특징 구현

엑셀 파일 파싱

엑셀 파일 파싱

```
public ExcelPs(String areaName, String sidoName) {
    this.areaName = areaName; //지역이름
    this.sidoName = sidoName; //시도이름
    System.out.println(sidoName);
    System.out.println(areaName);
    try {
        //엑셀파일의 경로
        FileInputStream inputStream = new FileInputStream("C:\\Users\\dbswo\\OneDrive\\바탕 화면\\code.xlsx");
        XSSFWorkbook workbook = new XSSFWorkbook(inputStream);
        // sheet 수 취득
        int sheetCn = workbook.getNumberOfSheets();
        System.out.println("sheet 수 : " + sheetCn);
        for (int cn = 0; cn < sheetCn; cn++) {
            System.out.println("취득하는 sheet 이름 : " + workbook.getSheetName(cn));
            System.out.println(workbook.getSheetName(cn) + " sheet 데이터 취득 시작");
            // 0번째 sheet 정보 취득
            XSSFSheet sheet = workbook.getSheetAt(cn);
            // 취득된 sheet에서 rows 수 취득
            int rows = sheet.getPhysicalNumberOfRows();
            System.out.println(workbook.getSheetName(cn) + " sheet의 row 수 : " + rows);
            // 취득된 row에서 취득대상 cell 수 취득
            int cells = sheet.getRow(cn).getPhysicalNumberOfCells(); //
            System.out.println(workbook.getSheetName(cn) + " sheet의 row에 취득대상 cell 수 : " + cells);
            for (int r = 0; r < rows; r++) {
                row = sheet.getRow(r); // row 가져오기
                if (row != null) {
                    for (int c = 0; c < cells; c++) {
                        cell = row.getCell(c);
                        sido = row.getCell(1);
                        if (cell != null) {
                            String value = null;
                            switch (cell.getCellType()) { //셀의 타입이
                                case FORMULA: //수식일 경우
                                    value = cell.getCellFormula();
                                    break;

```

```
                                case NUMERIC: //숫자일 경우
                                    value = "" + String.valueOf(new Double(cell.getNumericCellValue()).longValue());
                                    if(c==0)
                                        areaCodes = value;
                                    break;
                                case STRING: //문자일 경우
                                    value = "" + cell.getStringCellValue();
                                    if(sido.getStringCellValue().contains(sidoName)) { // 시 도
                                        if(c==3 && value.contains(areaName)) { //법정동
                                            areaCode = Long.parseLong(areaCodes)/100000; // 지역코드
                                        }
                                    }
                                    break;
                                case BLANK: //빈공간일 경우
                                    value = "[null 아닌 공백]";
                                    break;
                                case ERROR:
                                    value = "" + cell.getErrorCellValue();
                                    break;
                                default:
                                    }
                            } // System.out.print(value + "\t");
                        } else {
                            // System.out.print("[null]\t");
                        }
                    } // for(c) 끝
                } // System.out.print("\n");
            } // for(r) 끝
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```


04특징 구현

로그인

기능

사용자가 부동산 매물 정보제공 시스템을 사용할 수 있도록 사용자 정보를 받아 회원가입 후 프로그램을 사용할 수 있도록 함

요구사항

아이디와 비밀번호를 만들기 위해 개인의 정보를 필요하고 파일 입출력이 요구됨

구현 방안

텍스트 파일 입출력을 통해 작성한 양식을 텍스트 파일에 저장.
서버로 보내서 사용자가 로그인할 때 작성한 데이터와 비교하여 로그인 허가

04특징 구현

로그인

회원가입

```
okButton.addActionListener(e -> //OK 버튼을 눌렀을 때 이벤트 처리기
    if (!middlephoneNumber.getText().matches("[0-9]{4}$")) {
        JOptionPane.showMessageDialog(null, "폰번호 중간번호 오류 다시 입력하세요", "오류", JOptionPane.INFORMATION_MESSAGE);
    }
    else if (!endphoneNumber.getText().matches("[0-9]{4}$")) {
        JOptionPane.showMessageDialog(null, "폰번호 끝번호 오류 다시 입력하세요", "오류", JOptionPane.INFORMATION_MESSAGE);
    }
    else if (!emailField.getText().matches("[_a-z0-9-]+(.[_a-z0-9-]+)*@(?:\\w+\\.\\.)+\\w+$")) {
        JOptionPane.showMessageDialog(null, "이메일 형식 오류 다시 입력하세요", "오류", JOptionPane.INFORMATION_MESSAGE);
    }
    else if (String.valueOf(passwordField.getPassword()).equals(String.valueOf(passwordcheckField.getPassword()))) {
        //확인 버튼을 눌렀을 때 처리할 구문
        StringBuilder stringBuilder = new StringBuilder("회원가입");
        stringBuilder.append("\n");
        stringBuilder.append("이름 : ").append(nameField.getText()).append("\n");
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy년MM월dd일");
        String dateString = simpleDateFormat.format(birthday.getDate());
        stringBuilder.append("생년월일 : ").append(dateString).append("\n");
        stringBuilder.append("id : ").append(IDField.getText()).append("\n");
        stringBuilder.append("pw : ").append(String.valueOf(passwordField.getPassword())).append("\n");
        stringBuilder.append("휴대폰 번호").append(phone.getSelectedItemAt());
        stringBuilder.append("-").append(middlephoneNumber.getText());
        stringBuilder.append("-").append(endphoneNumber.getText()).append("\n");
        stringBuilder.append("email : ").append(emailField.getText()).append("\n");
        client.send(stringBuilder.toString());
        JOptionPane.showMessageDialog(null, "회원가입 성공! 로그인하세요~", "회원가입 성공!", JOptionPane.INFORMATION_MESSAGE);
        registFrame.dispose();
        registFrame.setVisible(false);
    }
    else {
        JOptionPane.showMessageDialog(null, "비밀번호를 다시 확인하세요", "비밀번호 오류", JOptionPane.INFORMATION_MESSAGE);
    }
});
```

04

특징 구현

로그인

로그인

```
loginButton.addActionListener(e -> {  
    String ID = idTextField.getText();  
    //getPassword()메소드는 char[] 즉 캐릭터 배열 형태  
    String password = String.valueOf(passwordField.getPassword());  
    StringBuilder stringBuilder = new StringBuilder("로그인");  
    stringBuilder.append("\n");  
    stringBuilder.append(ID).append("\n");  
    stringBuilder.append(password);  
    client.send(stringBuilder.toString());  
    //보내는 형식 - 로그인\n아이디\n비밀번호  
    //반환할 때 하기 위해서 String으로 바꿈  
    //ID와 PW에 idTextField, passwordField에 있는 문자열을 받아서 넣음  
});
```

04특징 구현

아이디/비밀번호 찾기

아이디 비밀번호 찾기!!

아이디 찾기 비밀번호 찾기

이메일로 찾기

이름 :

생년월일 : 12. 13.

이메일 :

찾기!

아이디 비밀번호 찾기!!

아이디 찾기 비밀번호 찾기

아이디 :

이름 :

생년월일 : 34. 12. 13.

찾기!

기능

사용자가 아이디와 비밀번호를 잊어버렸을 때 자신이 가입할 때 입력했던 정보와 현재 입력하는 정보를 비교하여 찾아줌

요구사항

아이디/비밀번호를 찾기 위해 입력한 데이터를 서버에 저장해야 하며, 파일 입출력이 요구됨

구현 방안

텍스트 파일 입출력을 통해 작성한 양식의 데이터를 텍스트 파일에 저장. 서버로 보낸 데이터와 사용자가 아이디/비밀번호를 찾을 때 작성한 데이터와 비교하여 로그인 관련 정보를 알려줌

04특징 구현

아이디/비밀번호 찾기

이메일로 아이디 찾기

```
private void idbutton() {
    groupBox3.removeAll();
    findIDPasswordDlg.remove(groupBox3);
    //이메일을 입력하면 아이디를 찾아준다.
    groupBox.setBorder(BorderFactory.createTitledBorder("이메일로 찾기"));
    gridBagConstraints.gridx = 0;
    gridBagConstraints.gridy = 1;
    gridBagConstraints.anchor = GridBagConstraints.WEST;
    findIDPasswordDlg.getContentPane().add(groupBox, gridBagConstraints);
    groupBox.setLayout(new GridBagLayout());
    //휴대폰번호를 입력하면 아이디를 찾아준다
    groupBox2.setBorder(BorderFactory.createTitledBorder("휴대폰번호로 찾기"));
    groupBox2.setLayout(new GridBagLayout());

    gridBagConstraints.gridx = 1;
    gridBagConstraints.gridy = 1;
    findIDPasswordDlg.getContentPane().add(groupBox2, gridBagConstraints);
    findByEmailsName.setText("이름 : ");
    gridBagConstraints.gridx = 0;
    gridBagConstraints.gridy = 0;
    groupBox.add(findByEmailsName, gridBagConstraints);
    gridBagConstraints.gridx = 1;
    findByEmailsNameField.setColumns(10);
```

```
groupBox.add(findByEmailsNameField, gridBagConstraints);
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 1;
findByEmailsBirth.setText("생년월일 : ");
groupBox.add(findByEmailsBirth, gridBagConstraints);
Calendar cal = Calendar.getInstance();
cal.add(Calendar.YEAR, -24);
findByEmailsBirthChooser.setDate(cal.getTime());
findByEmailsBirthChooser.setMaxSelectableDate(new Date());
gridBagConstraints.gridx = 1;
groupBox.add(findByEmailsBirthChooser, gridBagConstraints);
findByEmailsEmail.setText("이메일 : ");
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 2;
groupBox.add(findByEmailsEmail, gridBagConstraints);
gridBagConstraints.gridx = 1;
findByEmailsEmailTextField.setColumns(10);
groupBox.add(findByEmailsEmailTextField, gridBagConstraints);
findByEmailButton.setText("찾기!");
findByEmailButton.addActionListener(e -> {
    StringBuilder stringBuilder = new StringBuilder("이메일로 아이디 찾기");
    stringBuilder.append("\n");
    stringBuilder.append("이름 : ").append(findByEmailsNameField.getText()).append("\n");
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy년MM월dd일");
    String birthString = simpleDateFormat.format(findByEmailsBirthChooser.getDate());
    stringBuilder.append("생년월일 : ").append(birthString).append("\n");
    stringBuilder.append("이메일 : ").append(findByEmailsEmailTextField.getText()).append("\n");
    client.send(stringBuilder.toString());
});
```

04특징 구현

아이디/비밀번호 찾기

휴대폰 번호로 아이디 찾기

```
gridBagConstraints.gridwidth = 2;
gridBagConstraints.gridy = 3;
gridBagConstraints.gridx = 0;
groupBox.add(findByEmailButton, gridBagConstraints);
gridBagConstraints.gridwidth = 1;
findByPhoneNumberName.setText("이름 : ");
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 0;
groupBox2.add(findByPhoneNumberName, gridBagConstraints);
gridBagConstraints.gridx = 1;
findByPhoneNumberNameField.setColumns(10);
groupBox2.add(findByPhoneNumberNameField, gridBagConstraints);
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 1;
findByPhoneNumberBirth.setText("생년월일 : ");
groupBox2.add(findByPhoneNumberBirth, gridBagConstraints);
gridBagConstraints.gridx = 1;
groupBox2.add(findByPhoneNumberBirthChooser, gridBagConstraints);
findByPhoneNumberBirthChooser.setDate(cal.getTime());
findByPhoneNumberBirthChooser.setMaxSelectableDate(new Date());
gridBagConstraints.gridy = 2;
gridBagConstraints.gridx = 0;
```

```
findByPhoneNumberPhoneNumber.setText("휴대폰 번호 : ");
groupBox2.add(findByPhoneNumberPhoneNumber, gridBagConstraints);
gridBagConstraints.gridx = 1;
findByPhoneNumberPhoneNumberField.setColumns(10);
groupBox2.add(findByPhoneNumberPhoneNumberField, gridBagConstraints);
gridBagConstraints.gridwidth = 2;
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 3;
findByPhoneNumberButton.setText("찾기");
findByPhoneNumberButton.addActionListener(e -> {
    StringBuilder stringBuilder = new StringBuilder("휴대폰 번호로 아이디 찾기");
    stringBuilder.append("\n");
    stringBuilder.append("이름 : ").append(findByPhoneNumberNameField.getText()).append("\n");
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy년MM월dd일");
    String birthString = simpleDateFormat.format(findByPhoneNumberBirthChooser.getDate());
    stringBuilder.append("생년월일 : ").append(birthString).append("\n");
    stringBuilder.append("휴대폰 번호 : ").append(findByPhoneNumberPhoneNumberField.getText()).append("\n");
    client.send(stringBuilder.toString());
});
groupBox2.add(findByPhoneNumberButton, gridBagConstraints);
findIDPasswordDlg.getContentPane().revalidate();
findIDPasswordDlg.getContentPane().repaint();
}
```

04특징 구현

아이디/비밀번호 찾기

비밀번호 찾기

```
private void passwordbutton() {
    groupBox.removeAll();
    groupBox2.removeAll();
    findIDPasswordDlg.remove(groupBox);
    findIDPasswordDlg.remove(groupBox2);
    findIDPasswordDlg.getContentPane().revalidate();
    findIDPasswordDlg.getContentPane().repaint();
    gridBagConstraints.gridx = 0;
    gridBagConstraints.gridy = 1;
    gridBagConstraints.gridwidth = 2;
    findIDPasswordDlg.getContentPane().add(groupBox3, gridBagConstraints);
    gridBagConstraints.gridx = 0;
    gridBagConstraints.gridy = 0;
    gridBagConstraints.gridwidth = 1;
    JLabel id = new JLabel("아이디 : ");
    groupBox3.add(id, gridBagConstraints);
    gridBagConstraints.gridx = 1;
    JTextField idText = new JTextField();
    idText.setColumns(10);
    groupBox3.add(idText, gridBagConstraints);
    JLabel name = new JLabel("이름");
    gridBagConstraints.gridy = 1;
    gridBagConstraints.gridx = 0;
    groupBox3.add(name, gridBagConstraints);
    gridBagConstraints.gridx = 1;
    JTextField nameText = new JTextField();
    nameText.setColumns(10);
    groupBox3.add(nameText, gridBagConstraints);
    JLabel birth = new JLabel("생년월일");
```

```
    gridBagConstraints.gridy = 2;
    gridBagConstraints.gridx = 0;
    groupBox3.add(birth, gridBagConstraints);
    JDateChooser birthChooser = new JDateChooser();
    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.YEAR, -24);
    birthChooser.setDate(cal.getTime());
    birthChooser.setMaxSelectableDate(new Date());
    gridBagConstraints.gridx = 1;
    groupBox3.add(birthChooser, gridBagConstraints);
    gridBagConstraints.gridy=3;
    gridBagConstraints.gridx=0;
    gridBagConstraints.gridwidth=2;
    JButton findButton = new JButton("찾기!");
    groupBox3.add(findButton, gridBagConstraints);
    findButton.addActionListener(e -> {
        StringBuilder stringBuilder = new StringBuilder("비밀번호 찾기").append("\n");
        stringBuilder.append("아이디 : ").append(idText.getText()).append("\n");
        stringBuilder.append("이름 : ").append(nameText.getText()).append("\n");
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy년MM월dd일");
        String birthString = simpleDateFormat.format(birthChooser.getDate());
        stringBuilder.append("생년월일 : ").append(birthString).append("\n");
        client.send(stringBuilder.toString());
    });
    findIDPasswordDlg.getContentPane().revalidate();
    findIDPasswordDlg.getContentPane().repaint();
}

public void disposefindIDpasswordFrame(){
    findIDPasswordDlg.dispose();
    findIDPasswordDlg.setVisible(false);
}

public void showMessage(String message){
    JOptionPane.showMessageDialog(null, message, "응답", JOptionPane.INFORMATION_MESSAGE);
}
```

05 프로젝트 평가

프로젝트 완성도

구현

사용자 관리(로그인, 회원가입)

아이디/비밀번호 찾기

매물 위치 검색

건물 종류(아파트, 단독 다가구

등) 선택

GUI

70%

미구현

지도 매물 정보

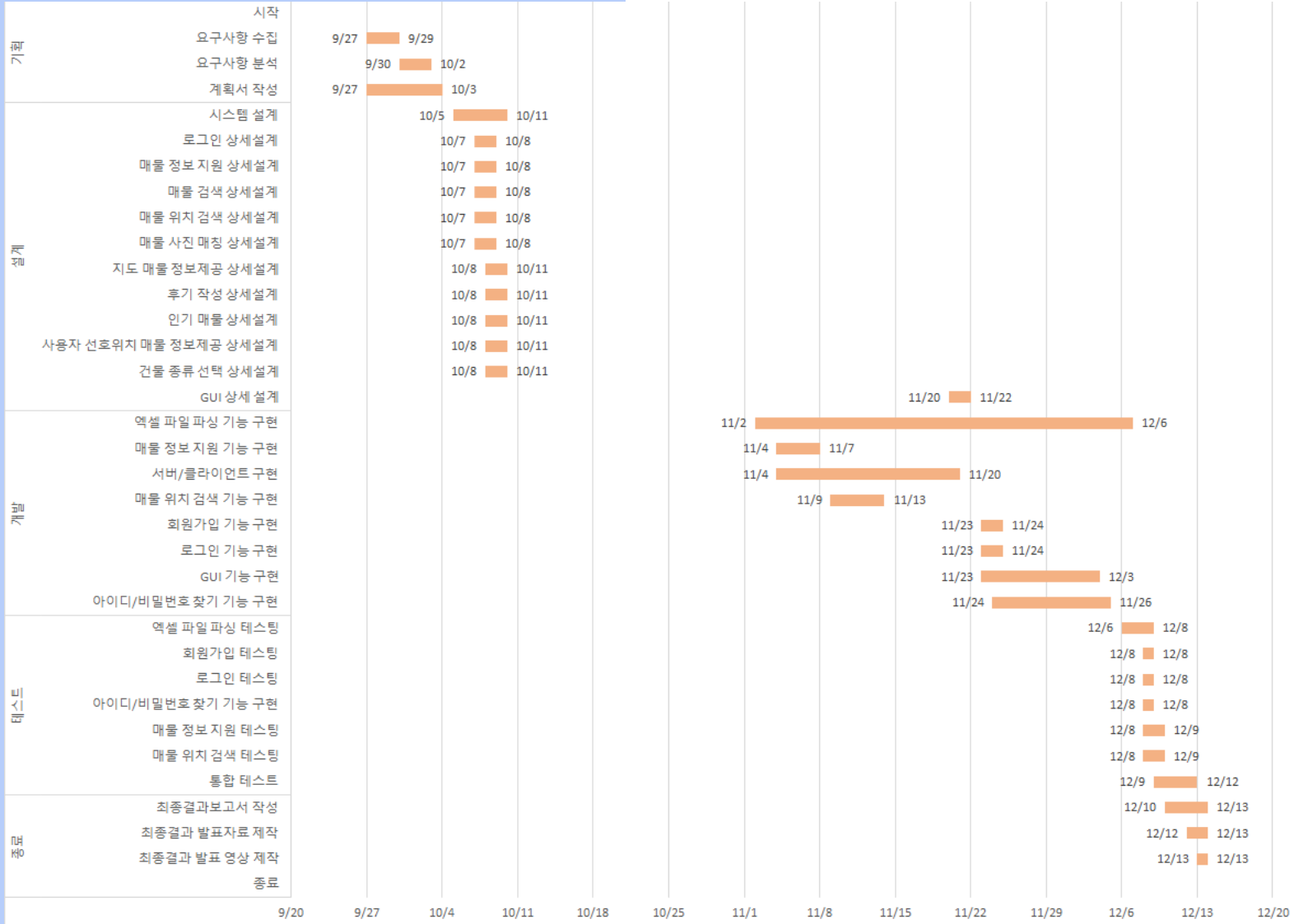
매물사진 매칭

사용자 선호위치 매물 정보

매물 찜하기

05

일정 계획 평가



05 프로젝트 평가

역할 수행 평가

작업	김준근	장윤재	김주의	김상환
아이디어 제공	√			
주제 선정	√	√	√	√
자료 조사			√	√
계획서 작성	√	√	√	√
회의록 작성		√	√	
엑셀 파일 작업	√	√		√
공공 API 분석	√	√	√	√
파싱	√	√		
코딩	√	√		
GUI 설계			√	√
GUI 구현	√	√		
테스팅		√	√	
형상 관리	√	√		
영상편집				√
최종 보고서	√	√	√	√
PPT			√	

05

프로젝트 평가

설계 구성요소

분석

시스템
설계

구현

05

프로젝트 평가

현실적 제한조건

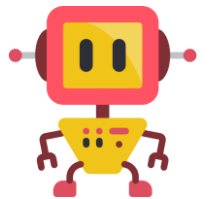
생산성

산업표준

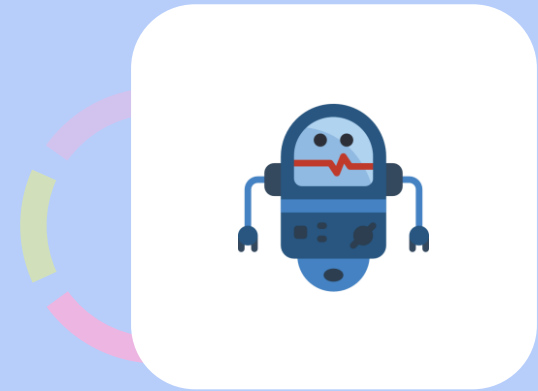
이번 학기 초부터 내주셨던 자바 팀 프로젝트는 내 손으로 시작되었다. 아무도 나서지 않아 내가 직접 나가 교수님께서 만드신 무작위 팀 만들기 프로그램을 행운의 숫자 7을 생각하여 7번을 돌려나온 조가 지금의 자바 1분반 7조이다. 조가 만들어졌을 때 다른 조와는 다르게 조원이 한 명 적은 명이라는 점에서 눈앞이 깜깜했다. '아 조원이 작아서 손해 보면 어쩌나, 열심히 하고자 하는 친구들이 한 명 줄어들 수도 있지 않을까?' 하며 걱정을 했다. 하지만 그것은 진짜 걱정에 불과했다. 우리 조원들은 나에게 팀장이 되어 달라고 해서 의도하지 않게 팀장이 되어 조원들에게 이것저것을 요구하고 부탁하는 저의 기대에 응해주며 우리 팀의 조원들은 한 명 적은 인원수에도 굴하지 않고 프로젝트를 같이 헤쳐나갔다. 먼저, 훌륭한 자바 실력을 겸한 준근이의 창의적인 코딩 머리와 뛰어난 임기응변 능력으로 우리 조의 부동산 프로그램을 한층 더 멋진 프로그램으로 완성시켜 주었고, 뛰어난 문서작성능력을 가진 주익은 여러 형식의 회의록과 보고서, 중간보고서, 소나큐브 보고서를 완벽하게 작성하고 미적인 부분까지 겸하여 정말 진짜 정말 문서를 잘 작성하였다. 상환이는 GUI를 그림으로 그려서 설계하고 프로그램이 완성되면 넣을 디자인 아이콘제작, 그리고 PPT를 보고 영상을 편집하는 능력을 보여주었다. 이



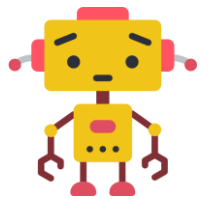
한 학기 동안 진행하였던 자바 팀 프로젝트는 다른 수업 팀 프로젝트와는 비교적 훨씬 길고 어려웠다. 처음 교수님이 팀을 무작위로 정해 주셨을 때 친하지 않은 조원 그리고 다른 조와는 다르게 조원이 한 명 적은 4명으로 구성된 팀이라 앞길이 막막했다. 하지만 팀원들과 프로젝트를 계획하고 수업시간에 배운 것을 토대로 역할을 분담하여 하나씩 건드려보니 기능이 구현되고 그때의 성취감은 말로 표현할 수가 없었다. 그렇게 팀원들과의 합도 점점 맞추고 프로젝트의 기능들을 하나씩 구현해 낼 때마다 자바라는 언어에 대해 이해할 수 있게 되었고 어느 순간 성장한 팀원들의 모습을 볼 수 있었다. 아쉬운 점은 다른 강의 과제와 시험 기간 때문에 시간이 부족하여 처음 계획했던 프로젝트를 100% 완성 시키지 못한 점이다. 그래도 군에서 전역하고 처음 하는 팀 프로젝트였는데 나름 성공적으로 끝내서 뿌듯하고 부족한 프로그래밍 실력이라도 나를 잘 이끌어준 팀원들에게 감사의 마음을 전하고 싶다.

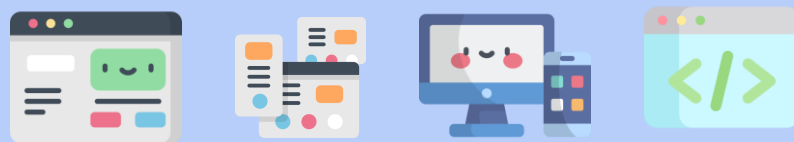


자바 팀 프로젝트는 굉장히 길고 어려운 프로젝트였다. 다른 수업과는 다르게 랜덤으로 팀원이 정해져 각기 다른 성격과 프로그래밍 실력을 가지고 있는 사람들과 약 3개월 동안 프로젝트를 진행 한다는 게 쉽지 않겠다고 생각이 들었다. 하지만 팀원 전체가 참여하려는 의지가 강하고 성격도 잘 맞아 순조롭게 프로젝트를 진행하였고 간간이 팀원들 간의 의견충돌도 있었지만 서로 배려하고 이해하며 프로젝트를 진행하였다. 처음 계획만큼 완벽하게 프로젝트를 완성 하진 못했지만 버려지는 팀원 없이 역할분담을 잘 하여 끝까지 프로젝트를 완성했다는 점에서 팀원들에게 크게 감동했다. 프로젝트가 끝날 때 즈음에는 자바라는 언어에 대해 좀 더 다가가 있는 나 자신을 보았고 이번 프로젝트가 팀 프로



힘들었다. 5명이 한 팀인데 우리 조는 4명이 팀이었고 다른 강의 과제들이 너무 많아서 힘들었다. 과제 때문에 하루종일 컴퓨터 보는 날이 많아져서 눈도 피로해지고 몸도 많이 안 좋아졌다. 그래도 상환이 오빠 빼고 다 아는 오빠들이어서 편했다. 원래 친하지 않은 사람에게는 말도 잘못하는데 아는 사람이니 불만이 생기면 바로 말할 수 있어서 좋았던 것 같다. 그리고 이 프로젝트로 몰랐던 상환이 오빠랑도 친해지고 다른 오빠들도 더 친해져서 좋았다. 그리고 제대로 된 프로젝트를 만들고 보고서를 쓰는게 진짜로 내가 프로그램을 만든 것처럼 느껴져서 좋았고 의미 있었다. 아쉬운 것은 이 프로젝트에서 나는 회의록, 중간/결과 보고서, 소나큐브 보고서 등 보고서들을 쓰는 게 중점이었어서 코딩은 오빠들이 거의 다 하고 나는 옆에서 피드백을 조금 해줬다. 개인 컴퓨터도 없어서 커밋도 못해보는 게 아쉬웠다. 그리고 다들 다른 강의 과제들도 하다 보니 시간이 부족해서 프로젝트를 미완성한 것이 아쉬웠다. 힘들었던 점은 회의록을 쓰는 게 처음이라 어떤 식으로 써야 할지 몰라서 막막했고 보고서 쓰는 것도 개념들을 많이 알아야 하는데 나는 모르는 게 너무 많아서 답답했다. 그래도 오빠들이 많이 도와주었고 역할분담을 해서 거기에 맞춰서 하니 나는 글 쓰는 것에 집중할 수 있어서 보고서와 글들을 쓸 수 있었던 것 같다. 프로젝트를 하면서 도움을 많이 받았고 다들 말하면 많이 도와줘서 고맙웠다. 다들





감사합니다.