

개발 완료 보고서

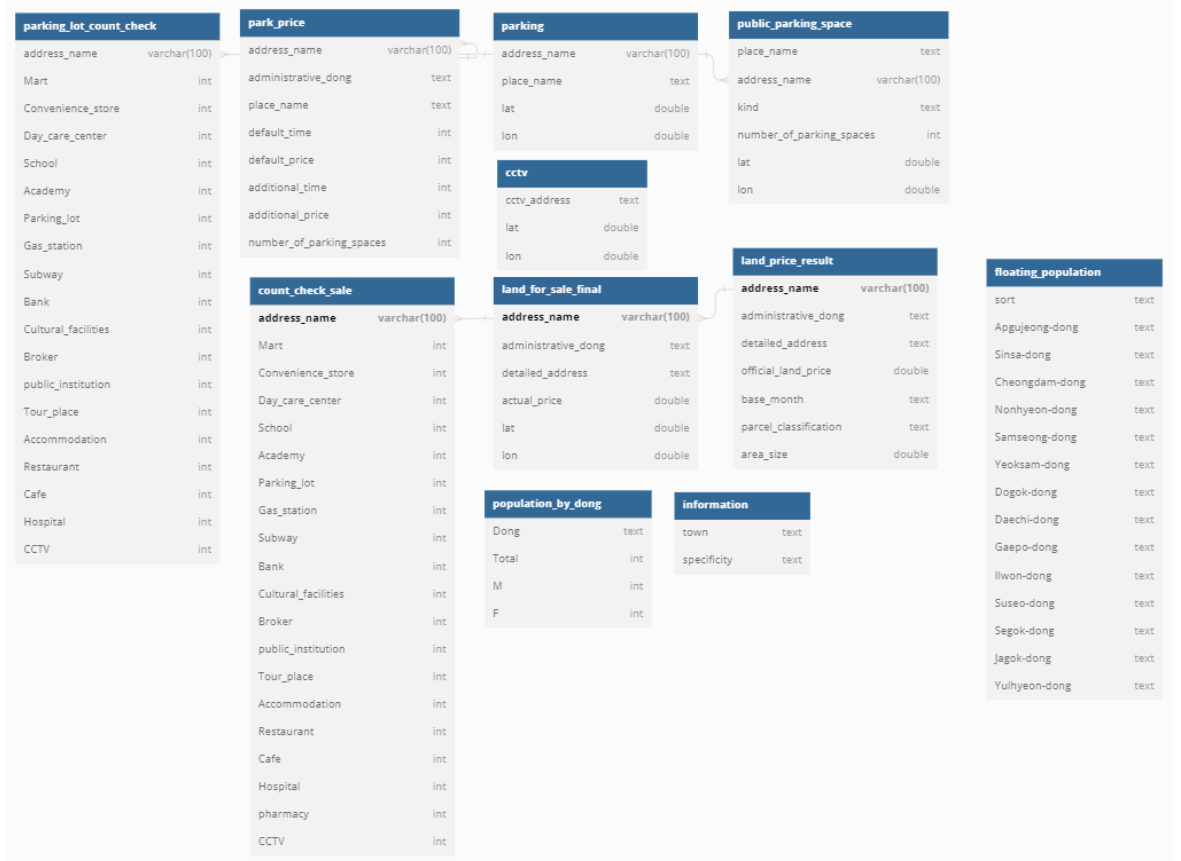
제출일 : 2023년 03월 12일

팀명	뉴파킹		
참여인원	이보라,김기태,이소윤,김명은		
프로젝트 소개			
프로젝트 명	데이터 분석을 통한 프랜차이즈 입지분석		
활동일시	2023.03.02 ~ 2023.03.10	장소	광주인력개발원 공학 1관 드론융합실
주요 주제	데이터 분석을 통한 강남구의 주차장 입지분석		
개발목적	<ul style="list-style-type: none">- 데이터 분석 과정을 경험하면서 데이터들의 상관관계를 찾고- 복잡한 문제에 대한 해결책을 찾기.- 원시데이터에서 유용한 인사이트 도출하기- TCP/IP기능을 C++로 구현해보기		
개인별 임무분담	이보라	기획 , 프로젝트 설계 , 데이터 수집 , 서버 기능 구현	
	김기태	DB 설계, 데이터 수집, 서버 기능 구현	
	이소윤	데이터 수집, 클라이언트 UI 설계 및 기능 구현	
	김명은	데이터 수집, 클라이언트 UI 설계 및 기능 구현	
개발환경	IDE : Pycharm, QT creator(CPP) / DBMS : MySQL / 공유환경 : GitHub		

일정표

필수												
실수실 땀근 시간 : 71시 30분 주석 잘 남기기(누가봐도 알아볼 수 있게 구체적으로) / 연주 이름은 정확하게 주가 코드 수정시 수정 날짜 및 수정사항 주석으로 적기												
일정표												
분류	항목		3/2		3/3		3/4		3/5		3/6	
학습	준비	구조도 작성/일정표/계획서										
		DB 작성										
	구현	서버 기능 구현										
		클라이언트 기능 구현										
		산출물 작성										
		제출										
분류	항목		3/8		3/9		3/10		3/11		3/12	
학습	준비	구조도 작성/일정표/계획서										
		DB 작성										
	구현	서버 기능 구현										
		클라이언트 기능 구현										
		산출물 작성										
		제출										
개인 일정												
이보라		특이사항 없음										
김기태		특이사항 없음										
김영은		특이사항 없음										
박소윤		특이사항 없음										

ERD



서버

코드 설명	void MainWindow::Read_Data_From_Socket() 	클라이언트가 보내준 데이터를 조건문으로 분기시켜준다.
	void MainWindow::Add_New_Client_Connection(QTcpSocket *socket) 	서버의 ip와 포트번호로 들어온 신규 클라이언트를 while문을 통해 Add_New_Client_Connection() 함수를 사용하여 연결합니다.
	void MainWindow::floting_population(QTcpSocket *socket) 	클라이언트가 요청한 데이터의 요청문구가 “유동인구”의 경우 json문서로 직렬화 시켜준다.
	void MainWindow::land_price(QTcpSocket *socket) 	클라이언트가 요청한 데이터의 요청문구가 “공시지가”의 경우 json문서로 직렬화 시켜준다.
	void MainWindow::parking_station(QTcpSocket *socket) 	클라이언트가 요청한 데이터의 요청문구가 “주차장개수”의 경우 json문서로 직렬화 시켜준다.
	void MainWindow::land_sale(QTcpSocket *socket,QString selected_address) 	클라이언트가 요청한 데이터의 요청문구가 “매물정보”의 경우 json문서로 직렬화 시켜준다.
	void MainWindow::search_score(QTcpSocket *socket, QString selected_address) 	클라이언트가 요청한 데이터의 요청문구가 “선택매물정보”의 경우 json문서로 직렬화 시켜준다.
	void MainWindow::sendlocation(QTcpSocket *socket,QString selected_address) 	클라이언트가 요청한 데이터의 요청문구가 “분석보기”의 경우 json문서로 직렬화 시켜준다.
	void MainWindow::send_json_object(QJsonDocument data,QTcpSocket *socket) 	클라이언트에게 json변환 데이터 보내주기
	클라이언트	

void MainWindow::Read_Data_From_Socket() 	서버에서 소켓을 통해 전송된 데이터를 읽어서 데이터의 크기를 계산한 후 데이터를 추출해준다. 보내준 데이터를 모두 읽어올 때까지 반복한다.
void MainWindow::processReceivedData(QByteArray receivedData) 	서버에서 보내준 json변환 데이터를 받는다. 조건문의 명령어와 서버에서 보내준 명령어가 일치하면 함수가 실행된다.
void MainWindow::on_comboBox_currentTextChanged(const QString &arg1) 	콤보박스 선택 시 텍스트를 받는다. 서버로 “매물정보” 명령어와 정보를 json 데이터로 보낸다.
void MainWindow::on_btn_enter_clicked() 	확인 버튼 누른 후 다음 페이지로 이동. 정보 라벨을 초기화한다. 주변 상점과 영업을익 그래프를 초기화한다. 서버로 “선택매물정보” 명령어와 정보를 json 데이터로 보낸다.
void MainWindow::on_btn_check_clicked() 	분석보기 버튼을 누르면 스택위젯이 변경돼서 지도와 그래프를 보여준다. 서버로 “분석보기” 명령어와 정보를 json 데이터로 보낸다.
void MainWindow::on_btn_back_clicked() 	이전 페이지로 이동한다.
void MainWindow::on_tableWidget_itemClicked(QTableWidgetItem *item) 	테이블에서 클릭한 행에서 주소를 받아와 변수로 저장해준다.
void MainWindow::see_location(QJsonObject obj) 	서버에서 받은 json 데이터의 위도와 경도로 주정차 단속위반 CCTV, 공영 주차장, 선택한 매물의 위치를 지도에 표시해준다.
void MainWindow::dong_location(QJsonObject obj) 	서버에서 받은 json 데이터의 위도와 경도로 콤보박스에서 선택한 행정동의 전체 매물의 위치를 지도에 표시해준다.
QString MainWindow::make_html(QString position_string, QString lat, QString lon) 	맵과 마커를 만드는 html 문서를 생성한다.

	void MainWindow::write(QString filename, QString positions_string, QString lat, QString lon)	파일에 데이터를 쓰는 함수.
	void MainWindow::view_chart(QJsonObject obj)	서버에서 받은 json 데이터의 값으로 선택한 매물의 주변 상점을 파이 차트로 생성한다.
	void MainWindow::view_chart2(QJsonObject obj)	서버에서 받은 json 데이터의 값으로 선택한 매물의 이익과 비용을 파이 차트로 생성한다.
	void MainWindow::Analysis_score(QJsonObject obj)	서버에서 받은 json 데이터의 값으로 주차대수, 월수익, 수익률과 점수를 보여준다.
	void MainWindow::on_search_btn_clicked()	다음 페이지로 이동한다.
	void MainWindow::on_comboBox_2_currentTextChanged(const QString &arg1)	유동인구, 평균공시지가, 주차장개수가 들어있는 콤보박스 안에 서버에서 보내준 json 데이터를 받는다.
	void MainWindow::ShowScatterChart(QJsonObject obj)	유동인구에 스캐터 차트를 생성하고 라벨로 1위부터 5위까지 순위를 나타낸다.
	void MainWindow::ShowScatterChart2(QJsonObject obj)	평균공시지가를 스캐터 차트를 생성하고 라벨로 1위부터 5위까지 순위를 나타낸다.
	void MainWindow::ShowScatterChart3(QJsonObject obj)	주차장개수를 스캐터 차트를 생성하고 라벨로 1위부터 5위까지 순위를 나타낸다.
개인 후기	이보라	파이썬으로 TCP / IP 네트워크 통신프로그램을 만드는 것과 차원이 달랐다. C++은 C++입니다. 하지만 QT라서 조금은 편했던 부분도 있었다. 따로 쓰레드를 사용하지 않아도 되는 부분과 UI를 만들때 go to slot으로 자동생성되는 헤더파일등 편리한 점들도 많았기 때문이다.

		<p>기업프로젝트인만큼 회의도 많이 진행했고 자료 분석과 요구분석도 더욱 꼼꼼하게 잡고 시작했습니다.</p> <p>크롤링하는 과정부터 데이터들을 분석하여 프로그램을 만들어 내는 과정 모든것들이 조금 더 개발자에 한걸음 다가선 느낌이 듭니다</p> <p>수집한 데이터들을 활용함으로써 가치있는 데이터로 사용할수있었습니다.</p> <p>데이터들을 분석하고 쪼개다보니 부족한부분들이 눈에 띄었고 정제화해준 DB담당자의 노고에 박수를 보냅니다.</p> <p>항상 느끼지만 새로운 프로젝트를 시작할때마다 새로운 언어에 도전하고 백지부터 점을 찍고 그림을 그려나가고 있다는 점들이 너무나 신기합니다.</p> <p>무에서 유를 창조할수 있도록 맡은 바 최선을 다해준 팀원들이 있었기에 마무리를 잘할수 있었습니다.</p>
	김기태	<p>파이썬 셀레니움을 이용해서 크롤링을 하는 과정에서 단순 반복작업을 다시 일일이 코드로 구현하여 매크로를 만들어내 크롤링을 하는과정이 신기했고 질이 좋은 자료이던 아니던 모두 긁어와서 정제하는 과정에서 많이 배웠습니다.</p> <p>또한 TCP/IP 서버와 클라이언트의 송신과정에 있어서 파이썬에서는 하지 않았던 데이터의 Json으로의 직렬화 과정및 소켓의 크기를 넘어서는 데이터의 전송을 해보면서 성취감을 느꼈습니다.</p> <p>제가 DB 설계를 맡았었는데 처음에 idx값등 테이블간의 관계를 생각하지 않고 PK,FK를 제대로 지정해주지 못하고 결국 그냥 주소 Text를 가지고 데이터를 찾게 되어 많이 아쉽습니다.</p> <p>그리고 QT의 WebEngineWidget을 사용하여 지도를 띄우면서</p> <p>카카오 API를 이용하여 이전에 배웠던 HTML 파일을 쓰고 그 데이터들의 좌표값을 지정하여 마커등을 넣어준다거나</p> <p>카카오 API를 통해 여러 카테고리의 장소등을 찾으면서와 이런 공개 API에도 많은 기능이 있구나 많이 써봐야겠다는 생각이 들었습니다.</p> <p>또한 좋은 팀장님과 팀원들을 만나서 기업프로젝트를</p>

		재밌게 그리고 다들 서로 잘 도와줘서 잘 마무리했습니다.
	이소윤	<p>첫번째 기업프로젝트여서 어떻게 하는 게 맞는지 고민이 많았습니다. 진행해가는 과정에서 팀원들과 회의를 많이 해서 방향성을 잡아나갔습니다.</p> <p>처음으로 API와 셀레니움을 이용해서 크롤링도 시도해보고 프로젝트의 주제에 맞는 데이터들을 많이 찾아보면서 데이터 수집과 분석의 어려움도 많이 느꼈습니다.</p> <p>C++ QT로 TCP/IP 서버와 클라이언트를 구현하는 것도 처음이어서 많이 막연했는데 조금씩 하다보니깐 구현이 돼서 성취감이 있었습니다. 파이썬과는 다르게 Json 파일을 주고 받을 때 처리해야 하는 부분들이 많아서 그 부분에서 어려움이 많았습니다. 그래프 구현과 지도를 띄우는 것도 이번 프로젝트를 통해서 처음해봐서 재미있었고 다음번에 한다면 더 잘 구현해내보고 싶습니다.</p> <p>처음으로 시도하는 게 많았던 이번 프로젝트였는데 팀원들과 함께 고민하고 해결해 나가면서 즐겁고 보람차게 마무리 할 수 있었습니다. 감사합니다.</p>
	김명은	<p>맨 처음 자료수집을 위해 크롤링 하는법도 신기했고 어떻게 하는지 모르겠고 어려웠습니다. 그래서 직접 자료수집을 하기도 했었는데 이번에 크롤링을 시도해봤으니 다음번에는 더 발전한 모습으로 잘 해야될 것 같습니다.</p> <p>자료수집을 하고 그 이후에 무엇을 해야할지 잘 모르겠었는데 자주 회의를 하여 역할분담을 한것이 좋았던 것 같습니다.</p> <p>C++로 서버와 클라이언트를 구현하는것도 처음이라 낯설었지만 그래도 같이 하다보니 조금은 이해되었던 것 같습니다.</p> <p>스캐터차트를 사용하면서 자료가 많이 없어서 막막했는데 구글에 검색하고 팀원들의 도움으로 조금씩 차트의 모습이 구현되는게 눈에보였습니다. 그래도 QT로 구현을 하게되어서 생각보다는 쉽게 접근할 수 있었습니다.</p> <p>그리고 C++을 이용한 프로젝트를 하라고 했을때</p>

		<p>기대도 되고 걱정이 있었지만 다행히 마칠 수 있어서 보람있었습니다. 팀원들과의 많은 회의와 도움으로 잘 마무리 되었던 것 같습니다.</p>
비고		