



Politechnika Śląska
Wydział Inżynierii Materiałowej

Programowanie i projektowanie aplikacji obiektowych

TEMAT:

**“Squad Węgla” - Aplikacja desktopowa do zarządzania wewnętrzną firmą
skupu węgla.**

Uwagi prowadzącego:

Data przyjęcia:

Podpis prowadzącego:

Nazwisko i Imię:

Bładziak Konrad

Czajkowski Michał

Kluczewski Stanisław

Kostyra Mateusz

Imię i nazwisko opiekuna:

dr inż. Adam Kachel

Grupa: IP30pp

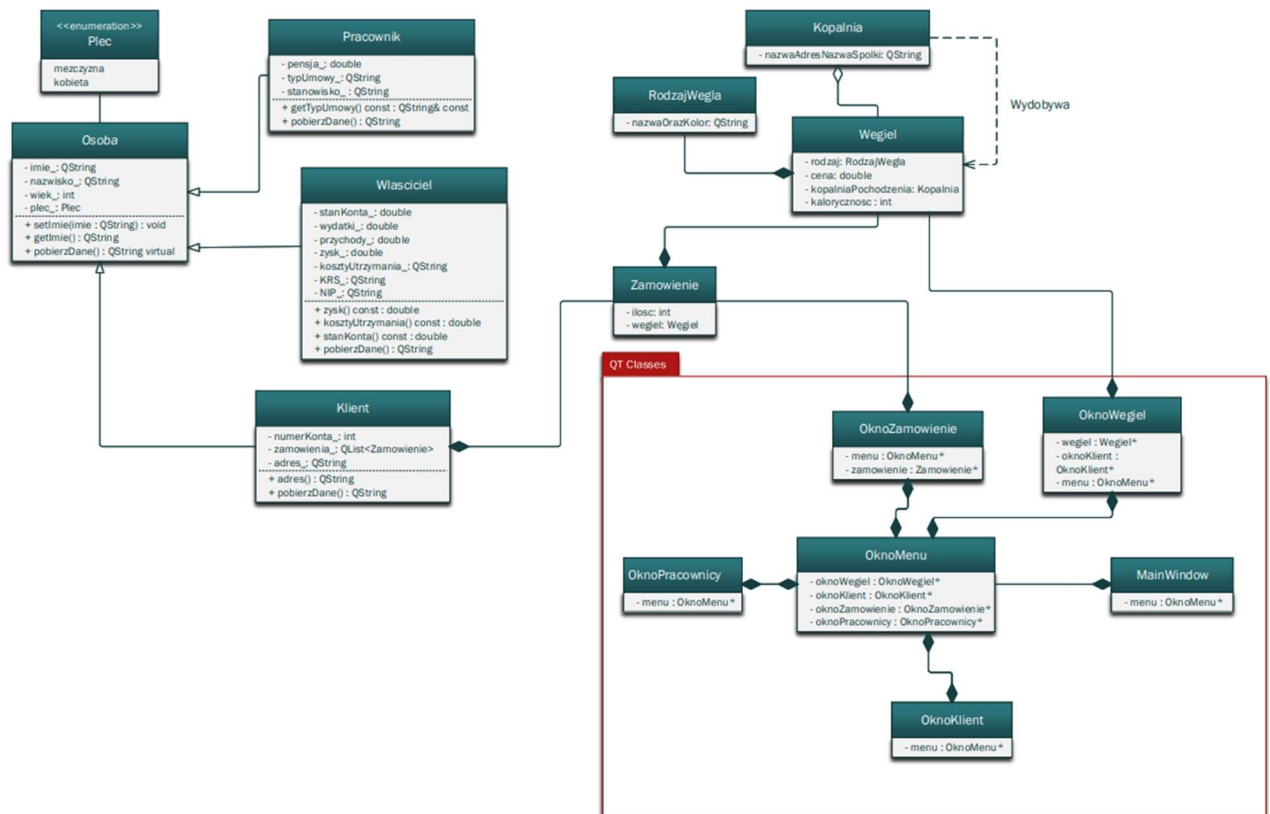
1. Opis aplikacji

Aplikacja jak i jej działanie ma na celu pomoc w prowadzeniu oraz inwentaryzacji jednoosobowej działalności gospodarczej - SquadWęgla(składnica węgla). Aplikacja posiada kilka okien, które odpowiednio służą do operacji CRUD na wybranych klasach (Węgiel, Osoba, Zamówienie).

2. Użyte technologie:

- QT
- SQLite
- GitHub

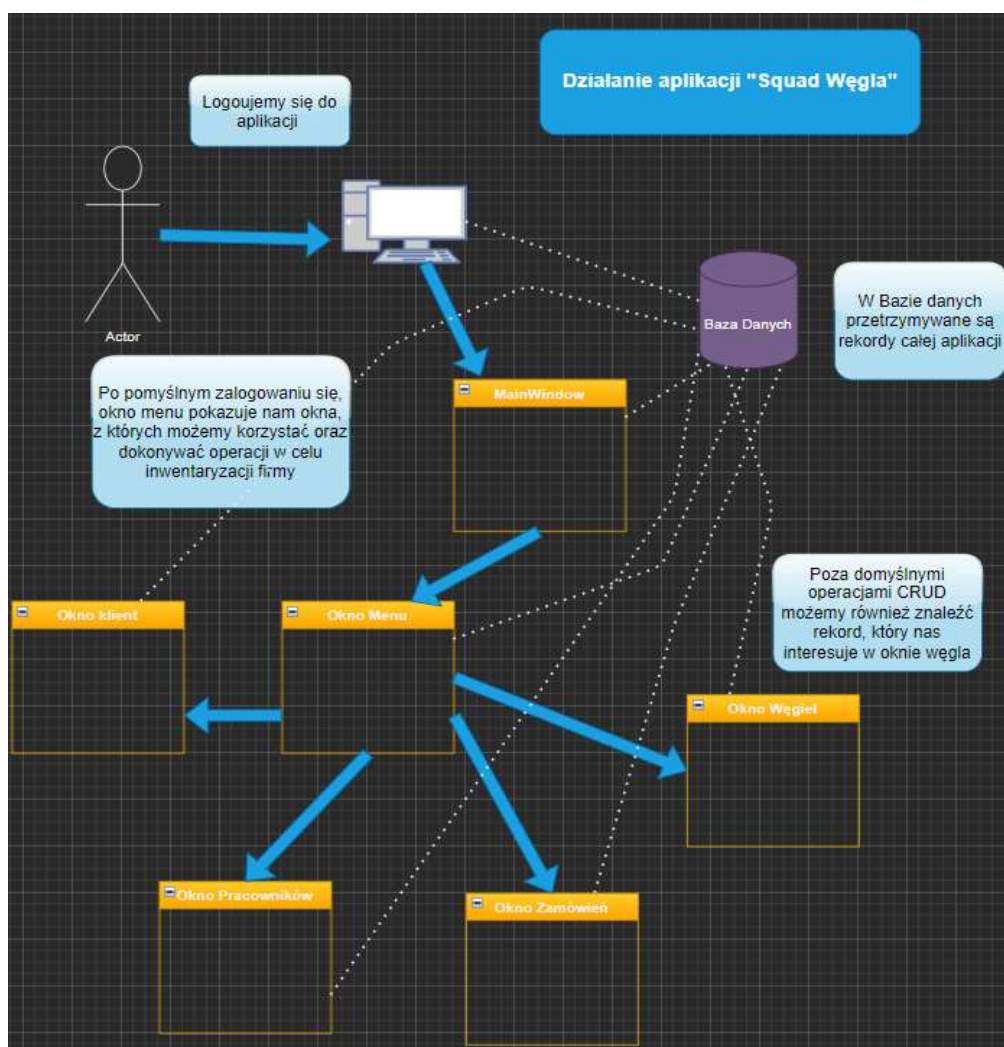
3. Schemat UML:



4. Schemat Bazy danych



5. Schemat działania aplikacji



6. Podział pracy oraz obowiązków

a) Konrad Bładziak

- Klasy
 - Osoba
 - Pracownik
 - Klient
- Programowanie działania okien umożliwiających funkcjonalności CRUD dla okien
 - Okno pracownicy
 - Okno klienci
 - Kontrola wersji git

b) Michał Czajkowski

- Klasy
 - Zamówienie
- Okno zamówienie – implementacja połączenia z bazą danych i metod CRUD dotyczących zamówień
- Schemat UML
- Drobne poprawki i ogólna pomoc przy innych problemach wymagających rozwiązania

c) Stanisław Kluczewski

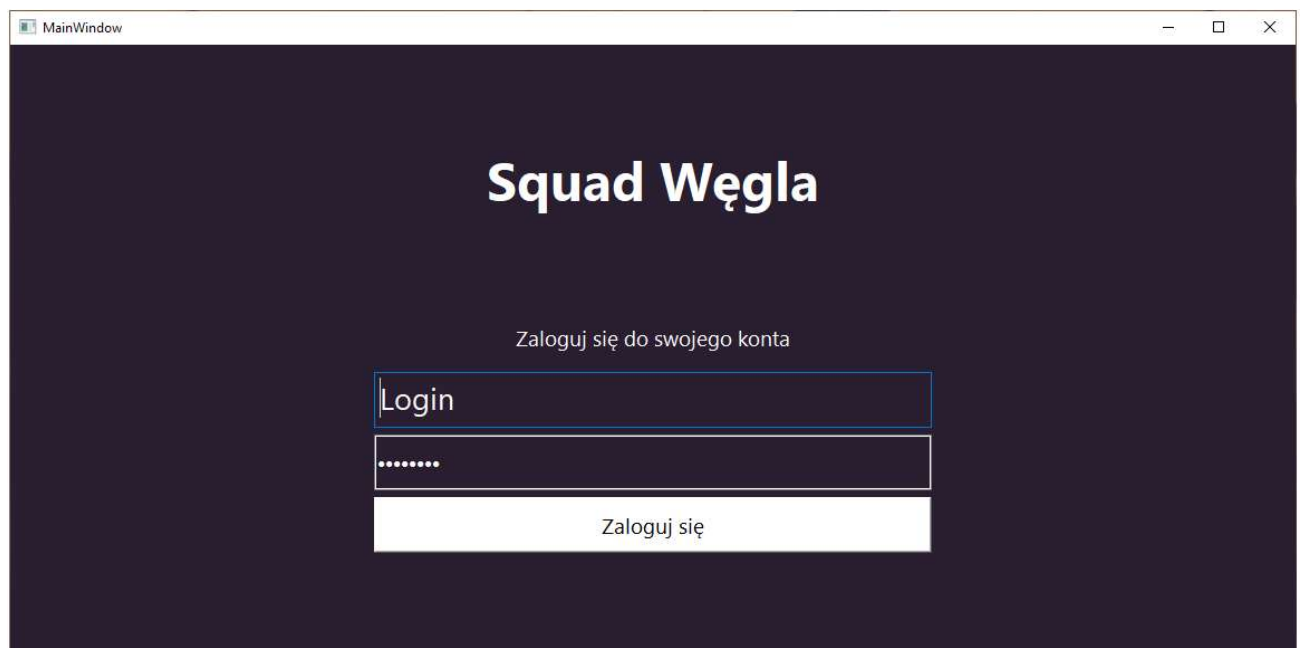
- Czuwanie nad funkcjonowaniem Back-End
- Klasy
 - Węgiel
 - Rodzaj Węgla
 - Kopalnia
- Okno Węgiel – czuwanie nad odpowiednim okna działaniem; implementacja poleceń oraz podłączenie do SQLite
- Realizacja okien, będących filarem operacji CRUD
- Schemat działania aplikacji

d) Mateusz Kostyra

- Odpowiedzialność za projektowanie okien(Front-End)
- Responsywność oraz wycentrowanie elementów UI
- Pomoc w realizacji klas oraz łączenia z bazą danych
- Dbanie o czytelność i przejrzystość kodu
- Schemat bazy danych

7. Wizualizacja okien

- Okno logowania



- Okno menu



- Okno klient

Dialog

Menu

Klient

Patryk Szymański 44 Mężczyzna Kolorowa 6,42-500 Będzin

Zofia Makówka 22 Kobieta Zielona 44,40-754 Katowice

Katarzyna Ogródek 37 Kobieta Plebiscytowa 33,40-035 Katowice

Sebastian Kowalski 23 Mężczyzna Ogródowa 58,00-876 Warszawa

Franciszek Nowak 36 Mężczyzna Robotnicza 37,40-750 Katowice

Sebastian Kowalski 23 Mężczyzna Ogródowa 58,00-876 Warszawa

Usuń rekord z bazy danych

Wybierz ID do usunięcia

Usuń

Dodaj klientów do bazy danych

Imię

Nazwisko

Adres

Wiek

Płeć Mężczyzna

Numer Konta

Dodaj

Edytuj klientów w bazie danych

Numer ID

Imię

Nazwisko

Adres

Wiek

Płeć

Numer Konta

Edytuj

- Okno węgiel

Dialog

Menu

Składnica Węgla

Wyświetl skład węgla

Znajdź po frazie

Szukaj

	Id	RodzajWęgla	Kopalnia	Cena	Kaloryczność
1	7	Rodzaj: Kamienny Kolor: Czarny	KWK Sośnica Błonie 6,44-103 ...	3200	31
2	8	Rodzaj: Kamienny Kolor: Czarny	KWK ROW Ruch Rydułtowy 44-2...	3000	35
3	9	Rodzaj: Kokosowy Kolor: Czarny	KWK Piast-Ziemowit 43-155 Bier...	1500	28
4	10	Rodzaj: Kamienny Kolor: Czarny	Zakład Górniczy Sobieski 43-600 ...	2600	30
5	11	Rodzaj: Brunatny Kolor: Czarno-...	KWB Bełchatów 97-400 Bełcható...	1300	34

Usuń rekord w bazie danych

Wybierz ID do usunięcia

Usuń

Dodaj nowy rekord do bazy danych

Kopalnia pochodzenia KWK Jastrzębie-Bzie 44-335 Jastrzębie Zdr

Rodzaj węgla Rodzaj: Brunatny Kolor: Czarno-Brunatny

Cena za tonę

Kaloryczność

Dodaj

Edytuj rekord w bazie danych

Numer ID

Kopalnia pochodzenia KWK Jastrzębie-Bzie 44-335 Jastrzębie Zdr

Rodzaj węgla Rodzaj: Brunatny Kolor: Czarno-Brunatny

Cena za tonę

Kaloryczność

Edytuj

- Okno pracownicy

Dialog

×

Powrót do menu

Pracownicy

Wyświetl Pracowników

	Id	Imie	Nazwisko	Wiek	login
1	1	adminImie	adminNazwisko	30	admin
2	2	Stanisław	Kluczewski	23	SKluczewski
3	3	Konrad	Bładziak	22	KBładziak
4	4	Mateusz	Kostyra	22	MKostyra
5	5	Michał	Czajkowski	22	MCzajkowski

Usuń rekord z bazy danych

Usuń

Dodaj pracowników do bazy danych

Imie

Nazwisko

Wiek

Dodaj

Edytuj pracowników w bazie danych

Id

Imie

Nazwisko

Wiek

Edytuj

- Okno zamówienia

Dialog

×

Menu

Zamówienia

Wyświetl zamówienia

	Id	RodzajWęgla	Ilość	Pracownik	Klient
1	9	Rodzaj: Kamienny Kolor: Czarny	2500	Imie: Stanisław Nazwisko: Kluczewski	Imie: Franciszek Nazwisko: Nowak adres: Robotnicza 37,40-750 Katowice
2	10	Rodzaj: Brunatny Kolor: Czarno-Brunatny	3000	Imie: Konrad Nazwisko: Bładziak	Imie: Franciszek Nazwisko: Nowak adres: Robotnicza 37,40-750 Katowice
3	11	Rodzaj: Koksowy Kolor: Czarny	5000	Imie: Mateusz Nazwisko: Kostyra	Imie: Sebastian Nazwisko: Kowalski adres: Ogrodowa 58,00-876 Warszawa
4	12	Rodzaj: Kamienny Kolor: Czarny	1500	Imie: Michał Nazwisko: Czajkowski	Imie: Katarzyna Nazwisko: Ogródek adres: Plebiscytowa 33,40-035 Katowice
5	13	Rodzaj: Kamienny Kolor: Czarny	2980	Imie: Stanisław Nazwisko: Kluczewski	Imie: Zofia Nazwisko: Makówka adres: Zielona 44,40-754 Katowice

Usuń rekord z bazy danych

Wybierz ID do usunięcia

Usuń

Dodaj nowe zamówienie do bazy danych

Klient

Pracownik

Rodzaj węgla

Ilość

Dodaj

Edytuj zamówienie

Numer ID

Pracownik

Klient

Rodzaj węgla

Ilość

Edytuj

8. Przykłady zrealizowanego kodu

- Osoba

```
enum Plec{
    mezczyzna,
    kobieta
};

class Osoba
{
private:
    QString imie_;
    QString nazwisko_;
    int wiek_;
    Plec plec_;

public:
    Osoba();
    Osoba(QString imie,QString nazwisko,int wiek,Plec plec);

    void setImie(QString imie);

    void setNazwisko(QString nazwisko);

    void setWiek(int wiek);

    void setPlec(Plec plec);

    QString getImie();

    QString getNazwisko();

    int getWiek();

    Plec getPlec();

    virtual QString pobierzDane();
};
```

- Rodzaj Węgla

```
class RodzajWęgla
{
private:
    QString nazwaOrazKolor;
public:
    // Konstruktory
    RodzajWęgla();
    RodzajWęgla(QString wszystko);
    void setNazwaOrazKolor(QString _nazwa);
    QString getNazwaOrazKolor();
};
```


- Właściciel

```
class Wlasciciel : public Osoba
{
private:
    double stanKonta_;
    double wydatki_;
    double przychody_;
    double zysk_;
    double kosztyUtrzymania_;
    QString KRS_;
    QString NIP_;

public:
    Wlasciciel();
    Wlasciciel(QString imie,QString nazwisko,int wiek,Plec plec,
                double stanKonta,double wydatki,double przychody,
                double zysk,double kosztyUtrzymania,QString KRS,QString NIP);

    double stanKonta() const;
    void setStanKonta(double stanKonta);
    double wydatki() const;
    void setWydatki(double wydatki);
    double przychody() const;
    void setPrzychody(double przychody);
    double zysk() const;
    void setZysk(double zysk);
    double kosztyUtrzymania() const;
    void setKosztyUtrzymania(double kosztyUtrzymania);

    const QString &KRS() const;
    void setKRS(const QString &KRS);
    const QString &NIP() const;
    void setNIP(const QString &NIP);

    QString pobierzDane() override;

};
```

- Klient

```
class Klient : public Osoba
{
private:
    int numerKonta_;
    QList<Zamowienie> zamowienia_;
    QString adres_;
public:
    Klient();
    Klient(QString imie,QString nazwisko,int wiek,Plec plec,int numerKonta,QString adres);

    void setNumerKonta(int numerKonta);

    void dodajZamowienie(Zamowienie zamowienie);

    void setAdres(QString adres);

    int getNumerKonta();

    QList<Zamowienie> getZamowienia();

    QString adres();

    QString pobierzDane() override;

};
```

- Pracownik

```
class Pracownik : public Osoba
{
private:
    double pensja_;
    QString typUmowy_;
    QString stanowisko_;

public:
    Pracownik();
    Pracownik(QString imie, QString nazwisko, int wiek, Plec plec, QString stanowisko, double pensja, QString typUmowy);

    void setPensja(double pensja);
    void setTypUmowy(const QString &newTypUmowy);

    double getPensja();
    const QString &getTypUmowy() const;

    QString pobierzDane() override;
};
```

- Węgiel

```
class Węgiel
{
private:
    RodzajWęgla rodzajWęgla;
    Kopalnia kopalniaPochodzenia;
    double cena;
    int kalorycznosc;
public:
    // Konstruktory
    Węgiel();
    Węgiel(RodzajWęgla _rodzaj, Kopalnia _kopalnia, double _cena, int _kalorycznosc);

    // Akcesory
    void setRodzajWęgla(RodzajWęgla _rodzaj);
    void setKopalniaPochodzenia(Kopalnia _kopalnia);
    void setCena(double _cena);
    void setKalorycznosc(int _kalorycznosc);
    RodzajWęgla getRodzajWęgla();
    Kopalnia getKopalniaPochodzenia();
    double getCena();
    int getKalorycznosc();
};
```

- Kopalnia

```
class Kopalnia
{
private:
    QString nazwaAdresNazwaSpolki;

public:
    // Konstruktory
    Kopalnia();
    Kopalnia(QString nazwaAdresNazwaSpolki);

    // Akcesory
    void setNazwaAdresNazwaSpolki(QString _nazwaAdresNazwaSpolki);
    QString getNazwaAdresNazwaSpolki();
};
```

- Zamówienia

```
class Zamowienie
{
private:
    int ilosc;
    Wegiel wegiel;

public:
    // Konstruktory
    Zamowienie();

    // Akcesory
    void setIlosc(int _ilosc);
    void setWegiel(Wegiel _wegiel);
    int getIlosc();
    Wegiel getWegiel();
};
```

9. Operacje CRUD wykorzystane w projekcie

Przykład działania okna, które odwzorowuje operacje CRUD (Create Read Update Delete).

```
this->db = QSqlDatabase::addDatabase("SQLITE");
this->db.setDatabaseName("../database.db");
```

Podłączenie do bazy danych – wykorzystujemy metodę statyczną z klasy **QSqlDatabase**, **addDatabase**, która przyjmuje jeden parametr - `connectionName`, w którym możemy przekazać informacje o np.: hoście, nazwie użytkownika, hasle użytkownika czy nazwie instancji bazy danych.

Dokumentacja: <https://doc.qt.io/qt-6/qsqldatabase.html#addDatabase>

Oto przykłady wykorzystania zapytania SQL w oknie działania Składnicy węgla:

a) Wyszukanie rekordu po frazie

QSqlQueryModel udostępnia model danych tylko do odczytu dla zestawów wyników SQL. Do poniższych przykładów użyjemy klasy QSqlQuery. Tworzymy instancje klasy, do której przypisujemy w metodzie prepare zapytanie select z SQL, która szuka rekordów o podanie frazie. Wyszukiwanie rekordu odbywa się poprzez użycie operatora like z języka SQL – szukamy rekordów, które spełniają ten warunek. W naszym przypadku szukamy wszystkich rekordów, które zawierają podaną treść w wewnątrz wiersza danej tabeli bazy danych (%phrase%). Metoda bindValue podaje pomocą dłoń w dopasowaniu szukanej wartości do miejsca gdzie chcemy ją szukać „:”. Metoda exec wywołuje tą zapytanie. Na koniec ustawiamy modal z wynikiem naszej bazy danych oraz ustawiamy szerokość kolumny.

Dokumentacja:

https://www.w3schools.com/sql/sql_like.asp

<https://doc.qt.io/qt-6/qsqlquery.html>

<https://doc.qt.io/qt-6/qsqlquerymodel.html#details>

https://www.w3schools.com/sql/sql_select.asp

```
void OknoWegiel::on_znajdzRekord_clicked()
{
    QString phrase = ui->searchRecord->toPlainText();
    phrase = "%" + phrase + "%";
    if(!db.open())
    {
        qDebug()<<"Error! Podłączenie do bazy danych nie powiodło się.";
    }
    else
    {
        QSqlQuery zapytanie;
        QSqlQueryModel *modal = new QSqlQueryModel();
        zapytanie.prepare("SELECT * FROM SkladnicaWegla WHERE RodzajWegla LIKE :phrase OR Kopalnia LIKE :phrase OR Cena LIKE :phrase OR Kalorycznosc LIKE :phrase");
        zapytanie.bindValue(":phrase", phrase);
        zapytanie.exec();
        ui->tableSkladnicaWegla->setModel(modal);
        modal->setQuery(zapytanie);
        ui->tableSkladnicaWegla->setModel(modal);
        ui->tableSkladnicaWegla->setColumnWidth(2, 300);
        ui->tableSkladnicaWegla->setColumnWidth(1, 300);
        db.close();
    }
}
```

b) Dodanie rekordu do składnicy węgla

Dodawanie (CREATE) rekordu odbywa się poprzez użycie instrukcji INSERT INTO z SQL. Na początku sprawdzamy, czy połączenie z bazą danych jest aktywne. Następnie przygotowujemy zapytanie i przy użyciu bindValue mapujemy pola danej tabeli z danymi wprowadzonymi przez użytkownika w oknie. Na końcu sprawdzamy, czy zapytanie zostało wykonane.

Dokumentacja:

https://www.w3schools.com/sql/sql_insert.asp

```

void OknoWegiel::on_dodajButton_clicked()
{
    RodzajWegla rodzaj(getRodzajWegla());
    Kopalnia kopalnia(getKopalniaWegla());
    this->wegiel->setCena(getCeneWegla());
    this->wegiel->setKalorycznosc(getKalorycznosc());
    this->wegiel->setRodzajWegla(rodzaj);
    this->wegiel->setKopalniaPochodzenia(kopalnia);

    if(!db.open())
    {
        qDebug() << "Error! Podłączenie do bazy danych nie powiodło się.";
    }
    else
    {
        QSqlQuery zapytanie;
        zapytanie.prepare("INSERT INTO SkladnicaWegla (RodzajWegla,Kopalnia,Cena,Kalorycznosc)"
                           "VALUES (:RodzajWegla, :Kopalnia, :Cena, :Kalorycznosc)");
        zapytanie.bindValue(":RodzajWegla", getRodzajWegla());
        zapytanie.bindValue(":Kopalnia", getKopalniaWegla());
        zapytanie.bindValue(":Cena", getCeneWegla());
        zapytanie.bindValue(":Kalorycznosc", getKalorycznosc());
        if(zapytanie.exec())
        {
            qDebug() << "Dodano rekord do bazy danych";
        }
        else
        {
            qDebug() << "Coś poszło nie tak!";
        }
        db.close();
    }
    on_wyswietlButton_clicked();
}

```

c) Wyświetlenie wszystkich rekordów składnicy węgla

Wyświetlenie (READ) wszystkich rekordów z bazy danych danej tabeli odbywa się poprzez użycie instrukcji select SQL. Zwracane są te pola, które są ważne i kluczowe do działania aplikacji.

```

void OknoWegiel::on_wyswietlButton_clicked()
{
    if(!db.open())
    {
        qDebug() << "Error! Podłączenie do bazy danych nie powiodło się.";
    }
    else
    {
        QSqlQueryModel *modal = new QSqlQueryModel();
        modal->clear();
        QSqlQuery zapytanie;
        zapytanie.prepare("Select Id, RodzajWegla, Kopalnia, Cena, Kalorycznosc from SkladnicaWegla");
        zapytanie.exec();
        modal->setQuery(zapytanie);
        ui->tableSkladnicaWegla->setModel(modal);
        ui->tableSkladnicaWegla->setColumnWidth(2, 300);
        ui->tableSkladnicaWegla->setColumnWidth(1, 300);
        db.close();
    }
}

```

d) Edycja rekordu w składnicy węgla

Aktualizacja (UPDATE), czyli operacja edycji odbywa się poprzez użycie instrukcji Update języka SQL. Najpierw wyszukiwany jest rekord o danym id, a następnie dopasowujemy pola z użyciem bindValue. Sprawdzamy czy zapytanie udało się zaktualizować a następnie zamykamy połączenie z bazą danych.

Dokumentacja:

https://www.w3schools.com/sql/sql_update.asp

```
void OknoWegiel::on_EdytujButton_clicked()
{
    int id = ui->EdytujID->toPlainText().toInt();
    QString kopalnia = ui->EdytujKrajPochodzenia->currentText();
    QString rodzaj = ui->EdytujRodzajWegla->currentText();
    double cena = ui->EdytujCena->toPlainText().toDouble();
    double kalorycznosc = ui->EdytujKalorycznosc->toPlainText().toDouble();

    if(!db.open())
    {
        qDebug() << "Error! Połączenie do bazy danych nie powiodło się.";
    }
    else
    {
        QSqlQuery zapytanie;
        zapytanie.prepare("UPDATE SkladnicaWegla SET RodzajWegla=:RodzajWegla, Kopalnia=:Kopalnia, Cena=:Cena, Kalorycznosc=:Kalorycznosc WHERE Id=:id;");
        zapytanie.bindValue(":id", id);
        zapytanie.bindValue(":RodzajWegla", rodzaj);
        zapytanie.bindValue(":Kopalnia", kopalnia);
        zapytanie.bindValue(":Cena", cena);
        zapytanie.bindValue(":Kalorycznosc", kalorycznosc);
        if(zapytanie.exec())
        {
            qDebug() << "Zaktualizowano rekord w bazie danych";
        }
        else
        {
            qDebug() << "Coś poszło nie tak!";
        }
        db.close();
    }
    on_wyswietlButton_clicked();
}
```


e) Usunięcie rekordu o podanym id

Najpierw tworzymy zmienną, do której pobierane jest ID rekordu podane przez użytkownika do usunięcia. Następnie połączenie z bazą jest otwierane oraz tworzona jest instancja klasy QSqlQuery służąca do zarządzania naszym zapytaniem. Potem zapytanie jest wykonywane i wysyłane. Na koniec połączenie jest zamykane, w konsoli wyświetla się informacja o obiekcie, który usunęliśmy oraz odświeżana jest lista wyświetlanych rekordów na ekranie.

Dokumentacja:

https://www.w3schools.com/sql/sql_delete.asp

```
void OknoWegiel::on_usunButton_clicked()
{
    QString index = ui->usunId->toPlainText();
    if(!db.open())
    {
        qDebug()<<"Error! Podłączenie do bazy danych nie powiodło się.";
    }
    else
    {
        QSqlQuery zapytanie;
        zapytanie.prepare("DELETE FROM SkladnicaWegla WHERE Id =?");
        zapytanie.addBindValue(index);
        zapytanie.exec();
        db.close();
        qDebug()<<"Usunąłeś obiekt o id: " + index;
    }
    on_wyswietlButton_clicked();
}
```