|  |  |  |  |
| --- | --- | --- | --- |
| **Wydział Matematyki Stosowanej**  **Politechnika Śląska** | | | |
| **Kierunek:** | | | Informatyka |
| **Stopień studiów:** | | | pierwszy |
| **Semestr:** | | | IV |
| **Przedmiot:** | | | Programowanie obiektowe i graficzne |
| **Prowadzący zajęcia:** | | | dr inż. Adam Zielonka |
| **Dokumentacja techniczna**  **realizacji projektu zespołowego** | | | |
| **Zespół projektowy** | | | |
| **Lider zespołu:** | | Tomasz Pierzchała (tomasz.pierzchala.7@gmail.com) | |
| **Wykonawca 1:** | | Kamil Kryus (kamilkryus@gmail.com) | |
| **Projekt** | | | |
| **Tytuł projektu:** | DB Editor | | |
| **Repozytorium:** | https://github.com/pierzchalatomasz/db-editor | | |
| **Cel projektu:** | Umożliwienie edycji baz danych poprzez interfejs graficzny bez konieczności znajomości języka MySQL. | | |
| **Opis projektu:** | Projekt ten umożliwia takie możliwości jak dodanie, edycję, usuwanie, przeglądanie nowych baz danych, tabel, kolumn, rekordów. | | |
| **Użyte narzędzia** | | | |
| Nazwa: | Visual Studio 2013 Professional | | |
| Licencja: | DreamSpark | | |
| Nazwa: | gitHub | | |
| Licencja |  | | |
| Nazwa: | Serwer MySQL | | |
| Licencja: |  | | |

**Historia zmian dokumentu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LP | Opis zmiany | Typ | Data zmiany | Autor | Uwagi |
| 1 | Przygotowanie szablonu dokumentacji technicznej, dodanie podstawowych informacji | D | 27.05.2016 | Kamil Kryus |  |
| 2 | Dodanie informacji o nowych klasach w projekcie, dodanie wstępnych opisów projektowych, dodanie szkiców projektowych, edycja instrukcji użytkownika | D,E | 18.06.2016 | Kamil Kryus |  |
| 3 | Opis funkcjonalności stanów, kontrolek, poprawa istniejących opisów | D,E | 19.06.2016 | Tomasz Pierzchała |  |
| 4 | Opis kolejnych stanów, ich funkcjonalności, działanie RecordsListingControl | D | 22.06.2016 | Tomasz Pierzchała |  |
| 5 | Dodanie kolejnych informacji z klas, dodanie kolejnej części instrukcji użytkownika | D | 25.06.2016 | Kamil Kryus |  |
| 6 | Dodanie nowych funkcjonalności programu, aktualizacja starych klas programu, aktualizacja funkcjonalności oprogramowania, aktualizacja instrukcji użytkownika | D,E | 27.06.2016 | Kamil Kryus |  |
| 7 | Dodanie nieopisanych funkcjonalności, korekta merytoryczna zdarzeń wysyłanych przez stany | D,E, M | 27.06.2016 | Tomasz Pierzchała |  |

Typ: E - edycja dokumentu, D - dodanie nowych treści, M - korekta merytoryczna

**1. Opis projektu**

*Obszerny i precyzyjny opis realizowanego projektu. Nietechniczna "ilustracja" tego czego projekt ma dotyczyć.*

Do pracy z nasza aplikacją wymagany jest uruchomiony serwer MySQL, z którym będziemy się mogli połączyć.

Po uruchomieniu aplikacji pojawi się okienko do logowania. Po wpisaniu danych (lub pozostawieniu domyślnych, wtedy zalogujemy się na konto administratora) i naciśnięciu przycisku Connect, przejdziemy do głównego okna aplikacji. Po lewej stronie powinna pojawić się lista baz danych z jakimi użytkownik jest w stanie pracować. W dowolnym momencie może dodać nową bazę danych, gdyż w dolnym lewym rogu jest miejsce na wprowadzenie nowej nazwy bazy danych i zatwierdzenie tej operacji. W międzyczasie prawa strona programu pozostaje pusta.

Gdy poprzez kliknięcie wybierzemy bazę danych, na jakiej chcemy pracować (można ją zmienić w dowolnym momencie), to po prawej stronie pojawi się lista obecnych tabel w danej bazie danych. W dolnym prawym rogu możemy zawsze dodać nową tabelę do bazy klikając w odpowiedni przycisk.

Każda z tabel zawiera 3 możliwe operacje:

1)”Edit”

Wyświetli nam listę pól (kolumn) danej tabeli wraz z wypełnionymi ich właściwościami na kontrolce. Właściwości te można zmieniać, aplikacja pilnuje pracy użytkownika by nie dokonywał niedozwolonych operacji. Ewentualną zmianę w danym polu zatwierdzamy przyciskiem Save.

W każdej chwili również możemy usunąć daną kolumnę naciskając przycisk „Delete”. Pojawi się wtedy pytanie o pewność wykonania tej operacji.

Aby dodać nową kolumnę do tabeli wystarczy w dole aplikacji nacisnąć „Add new field” i odpowiednio wypełnić daną kontrolkę.

2)”Records”

Wyświetli nam listę pierwszych pięćdziesięciu rekordów w danej tabeli. Przeglądanie dalszych (stron) rekordów umożliwia nam przycisk Next lub Prev.

Chcąc usunąć lub edytować dany rekord, najpierw należy go wybrać, a następnie nacisnąć odpowiedni przycisk.

Edycja polega na: zmianie wartości pól w kolumnach, które się pojawią i kliknięciu przycisku Save.

W każdej chwili jesteśmy w stanie dodać nowy rekord poprzez przycisk „Add new record” w dolnym prawym rogu programu. Pojawią się wtedy kontrolki wraz z miejscami do wypełnienia.

3)”Delete”

Przycisk ten upewnieniu się usuwa daną tabelę z bazy danych.

**2. Wymagania**

Członkowie zespołu prowadząc wspólnie analizę zebranych wymagań dotyczących realizacji projektu pogrupowali je w następujący sposób:

2.1 Wymagania funkcyjne

-tworzenie nowych baz danych,

-usuwanie baz danych,

-zmiana nazwy bazy danych,

-tworzenie nowych tabel,

-usuwanie tabel,

-zmiana nazwy tabel,

-tworzenie nowych kolumn,

-usuwanie kolumn,

-zmiana nazwy lub właściwości kolumn,

-tworzenie nowych rekordów,

-usuwanie rekordów,

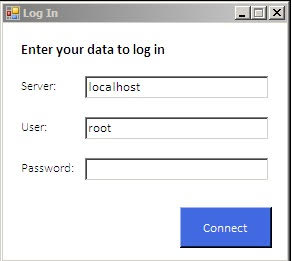
-zmiana poszczególnych lub wszystkich wartości w rekordzie.

2.2 Wymagania niefunkcyjne

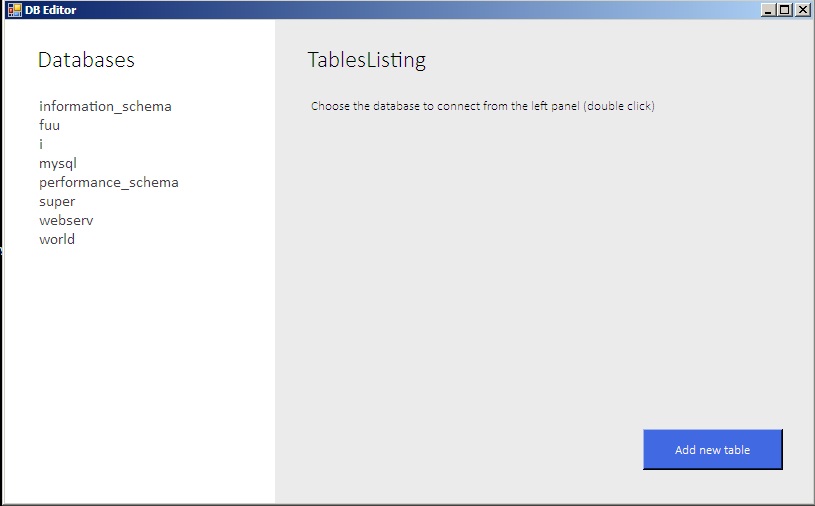
**3 Projekt interfejsu graficznego**

*Szkice wyglądu interfejsu publicznego aplikacji w odniesieniu do zebranych wymagań*

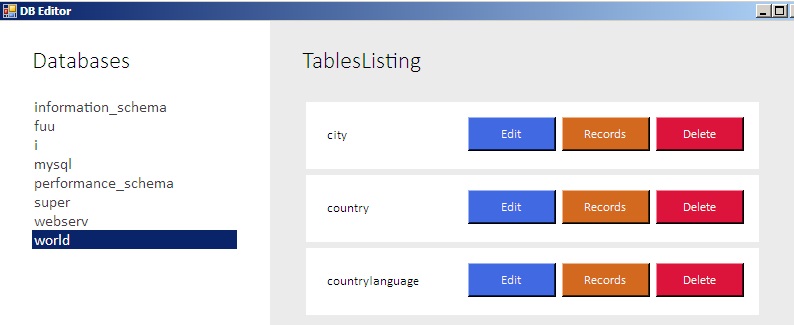
*Główne okno do logowania:*

**

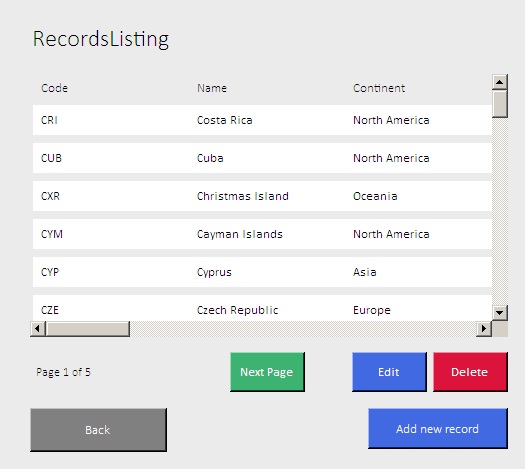
*Główne okno po zalogowaniu się użytkownika:*

**

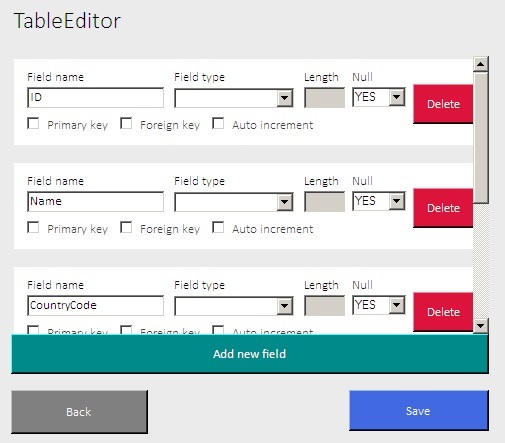
*Lista tabel po wybraniu bazy danych:*

**

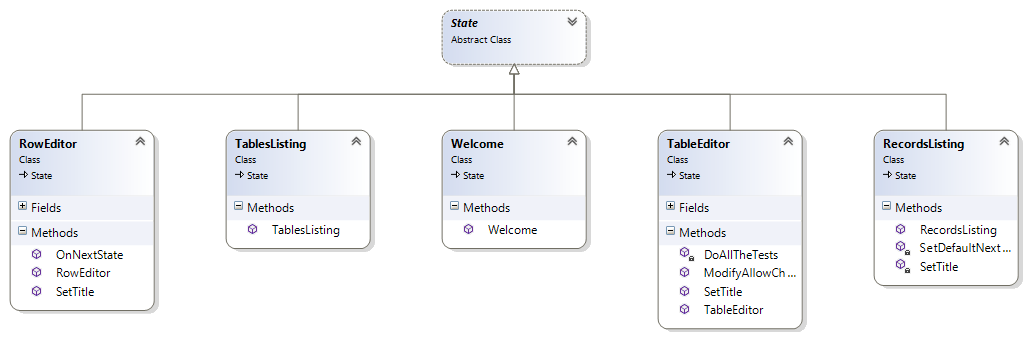
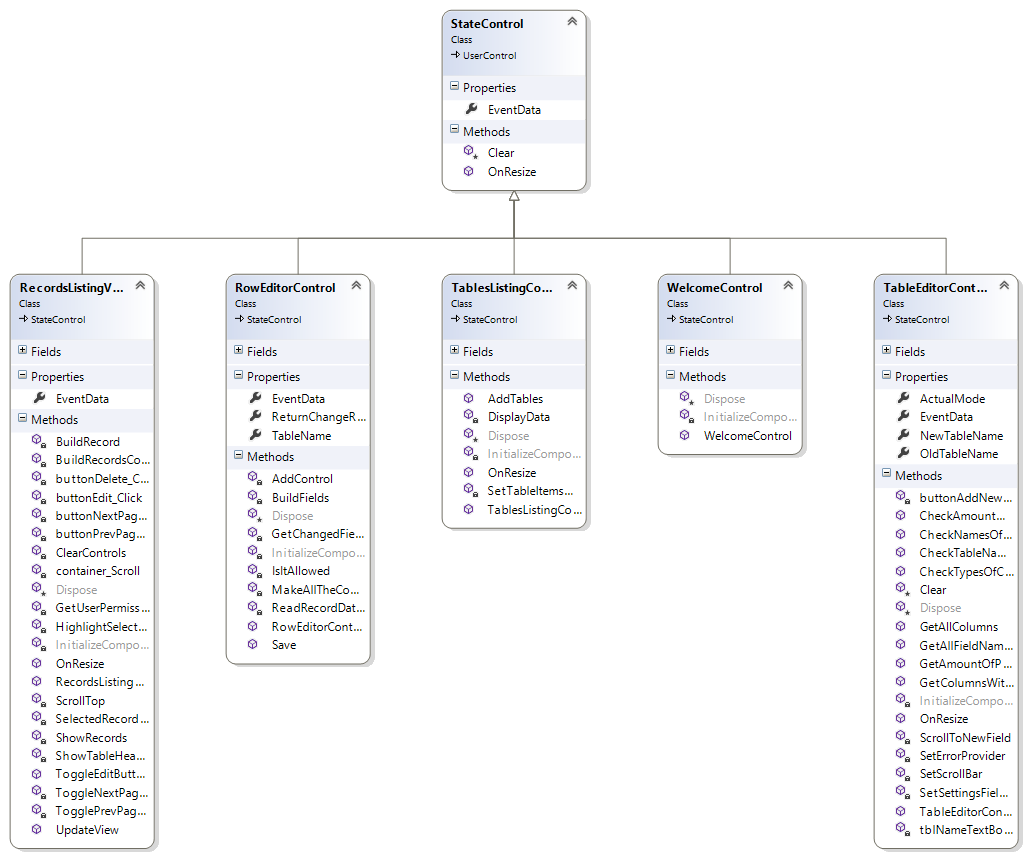
*Lista rekordów po wybraniu tabeli:*

**

*Możliwość edycji pół:*

**

**4. Model obiektowy**



**5. Model danych (w przypadku aplikacji bazodanowej)**

Aplikacja ma mieć możliwość edycji każdej MySQL’owej bazy danych.

***6. Implementacja***

*Dla każdej z klas opisujemy dokładnie jej interfejs publiczny*

***6.1 Klasy obsługujące połączenia z bazą danych***

|  |  |
| --- | --- |
| **Klasa** | NazwaKlasy |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
|  |  |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |

|  |  |
| --- | --- |
| **Klasa statyczna** | Database |
| **Właściwości** | |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| OperationResult CreateDatabase | *Za pomocą komendy MySQL tworzy bazę danych o podanej nazwie (przyjętej jako argument). Zwraca strukturę OperationResult informującą o wyniku operacji oraz ewentualnym błędzie.* |
| OperationResult DropDatabase | *Usuwa aktualnie używaną bazę danych. Zwraca strukturę OperationResult* |
| OperationResult DropDatabase | *Usuwa bazę danych o podanej nazwie (jako argument). Zwraca strukturę OperationResult.* |
| OperationResult RenameDatabase | *Metoda ta tworzy bazę danych o podanej nazwie, przepisuje wszystkie tabele ze wcześniejszej bazy danych i usuwa starą bazę danych. Zwraca strukturę OperationResult.* |
| OperationResult CreateTable | *Tworzy tabelę o podanej nazwie tworząc przy tym kolumny zadane w argumencie oraz przypisuje klucz główny. Zwraca strukturę OperationResult.* |
| OperationResult CreateTableWithMultiplePrimaryKeys | *Tworzy tabelę o podanej nazwie tworząc przy tym kolumny zadane w argumencie oraz przypisuje kolumny do klucza głównego. Zwraca strukturę OperationResult.* |
| OperationResult DropTable | *Usuwa tabelę o podanej nazwie w bazie danych. Zwraca strukturę OperationResult.* |
| OperationResult RenameTable | *Zmienia starą nazwę tabeli na nową podaną w argumencie. Zwraca strukturę OperationResult.* |
| List<string> GetDatabases | *Metoda ta wykonuje zapytanie do bazy danych o listę baz danych. Zwraca ich nazwy w postaci listy nazw.* |
| List<string> GetTablesFromDatabase | *Wykonuje zapytanie do bazy danych o listę nazw tabel w oparciu o nazwę bazy danych. Zwraca listę nazw tabel.* |
| List<string> GetFieldNamesFromTable | *Wykonuje zapytanie do bazy danych o nazwę pól z danej tabeli. Zwraca listę kolejnych nazw pól.* |
| List<ColumnStructureCreator> GetColumnSctuctureCreatorsFromTable | *Wykonuje zapytanie do bazy danych i na podstawie zwróconego wyniku tworzy listę kolumn przedstawianych jako obiekty ColumnStructureCreator. Zwraca ich listę.* |

|  |  |
| --- | --- |
| **Klasa statyczna** | Table |
| **Właściwości** | |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| OperationResult AddColumn | *Za pomocą komendy MySQL dodaje kolumnę do podanej tabeli (przyjętej jako argument) o konkretnej nazwie i właściwościach. Zwraca strukturę OperationResult informującą o wyniku operacji oraz ewentualnym błędzie.* |
| OperationResult ChangeColumn | *Zmienia kolumnę zadaną jako argument na inną (również podaną jako argument). Zwraca strukturę OperationResult.* |
| OperationResult DropColumn | *Usuwa kolumnę w tabeli podanej jako argument. Zwraca strukturę OperationResult.* |
| OperationResult MoveColumn | *Zmienia pozycję kolumny na podaną. Zwraca strukturę OperationResult.* |
| OperationResult AddPrimaryKey | *Przypisuje konkretnej kolumnie klucz główny w tabeli. Zwraca strukturę OperationResult.* |
| OperationResult DropPrimaryKey | *Przystosowuje podaną kolumnę jako możliwą do usunięcia klucza głównego, następnie usuwa klucz główny w tabeli. Zwraca strukturę OperationResult.* |
| OperationResult AddForeignKey | *Przypisuje klucz obcy w tabeli jako referencję do innej kolumny w innej tabeli. Zwraca strukturę OperationResult.* |
| OperationResult DropForeignKey | *Usuwa konieczny wpis w tabeli* information\_schema.key\_column\_usage, następnie usuwa klucz obcy w tabeli. *Zwraca strukturę OperationResult.* |
| Int GetAmountOfPages | *Wykonuje zapytanie do bazy danych o ilość rekordów w tabeli i poprzez działanie matematyczne zwraca ilość stron, jaka będzie potrzebna by wyświetlić wszystkie rekordy.* |
| List<List<string>> GetPageOfRecordsByIndex | *Wykonuje zapytanie do bazy danych o pięćdziesiąt rekordów zaczynając od n-tego, który jest podawany jako argument. Zwraca listę, która zawiera listę poszczególnych wartości pól w rekordzie.* |
| String GetFieldDataType | *Wykonuje zapytanie do bazy danych o typ danych danej kolumny.* |
| String[] GetEnumValues | *Wykonuje zapytanie do bazy danych o wartości, jakie może przyjąć dana kolumna, jeżeli jest typem enum. Następnie poprzez operacje na ciągach znaków zwraca listę możliwych wartości.* |

|  |  |
| --- | --- |
| **Klasa statyczna** | Record |
| **Właściwości** | |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| OperationResult ChangeRowValue | *Za pomocą komendy MySQL zmienia wartości wpisane w konkretnym wierszu rekordów w tabeli (przyjętych jako argumenty). Zwraca strukturę OperationResult informującą o wyniku operacji oraz ewentualnym błędzie.* |
| OperationResult InsertRowValue | *Przy pomocy zapytania do bazy danych tworzy rekord w danej tabeli. W zapytaniu podajemy nazwę kolumn, których używać będziemy (dostajemy to jako argument) oraz wartości pól. Zwraca strukturę OperationResult.* |
| Int GetNextIndex | *W przypadku, gdy pole jest typu auto\_increment metoda ta zwraca następny index, jaki powinien zostać dołączony do pola o takiej właściwości.* |
| OperationResult DeleteRow | *Przy pomocy zapytania do bazy danych usuwa konkretny wiersz danych. Zwraca strukturę OperationResult.* |

|  |  |
| --- | --- |
| **Klasa** | ColumnStructureCreator |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| string Field | *Reprezentuje nazwę kolumny.* |
| string Type | *Reprezentuje typ kolumny.* |
| String TypeLength | *Reprezentuje długość typu danej kolumny, jeżeli jest to konieczne.* |
| bool NullValue | *Reprezentuje możliwość dodawania pustych wartości do tej kolumny. W przypadku ustawienia true, występuje możliwość dodania pustej wartości do kolumny. W przypadku gdy występuje klucz główny/obcy, wartość tej właściwości musi być false.* |
| bool Primary\_Key | *Reprezentuje informację, czy dana kolumna jest kluczem głównym. True oznacza iż jest.* |
| string Default | *Reprezentuje domyślną wartość kolumny, jeżeli nie zostanie podana. Niemożliwe jest podanie domyślnej wartości, gdy właściwość Extra jest ustawiona na true.* |
| bool Extra | *Reprezentuje posiadanie cechy auto inkrementacji danej kolumny. Gdy ustawiona na true, kolumna posiada tą cechę. Gdy ustawiona na true, wartość właściwości Primary\_Key ustawiona na true, a Default i NullValue na false.* |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| override string ToString | *Nadpisana metoda zwracająca string, który reprezentuje kolumnę poprzez ustawione jej właściwości.* |
| String ToStringWIthoutPrimaryKeys | *Metoda, która zwraca ciąg znaków z właściwości danego obiektu ColumnStructureCreator jeżeli wywoływana jest metoda z kilkoma kolumnami należącymi do klucza głównego.* |
| Static bool operator == | *Porównuje ze sobą dwa obiekty ColumnStructureCreator pod względem właściwości i zwraca prawdę, jeżeli są takie same.* |
| Static bool operator != | *Wywołuje operator porównania na dwóch obiektach ColumnStructureCreator, jeżeli są takie same, to zwraca fałsz.* |

|  |  |
| --- | --- |
| **Struktura** | OperationResult |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| bool IsSucceded | *Reprezentuje pomyślność wykonania zapytania do bazy danych. Wartość true oznacza poprawne wykonanie zapytania. Niemożliwe jest ustawienie jej wartości poza konstruktorem.* |
| Exception Exception | *Reprezentuje wyjątek w przypadku błędu zapytania do bazy danych. Pozwala wyciągnąć informację o przyczynie błędu i poinformowaniu użytkownika.*  *Niemożliwe jest ustawienie jej wartości poza konstruktorem.* |

|  |  |
| --- | --- |
| **Klasa statyczna** | DBConnectionManager |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Static MySqlConnection Connection | Zwraca połączenie z bazą danych. |
| Static string ServerUrl | Pozwala ustalić adres serwera. |
| Static string Username | Pozwala pobrać i ustalić aktualnego użytkownika. |
| Static Password | Pozwala ustalić hasło. |
| Static DatabaseName | Pozwala ustalić i pobrać aktualnie używaną bazę danych. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Static void Connect | *Tworzy połączenie z bazą danych.* |
| Static QueryResult Query | *Wykonuje zapytanie do bazy danych, które wymaga zwracania wyników. Zwraca Query Result, który jest aliasem dla List<Dictionary<string, string>>* |
| Static bool IsConnected | *Zwraca informację, czy połączenie z bazą danych jest aktywne.* |
| Static void TryLogIn | *Sprawdza czy użytkownik może zalogować się na serwer MySQL.* |

***6.2 Klasy odpowiedzialne za uwierzytelnianie***

|  |  |
| --- | --- |
| **Klasa** | View |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

***6.3 Klasy odpowiedzialne za główne okno do operacji***

***6.3.1 Definicje***

**DEFINITIONS**

|  |  |
| --- | --- |
| **Klasa** | State |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| StateControl Control | Zwraca kontrolkę, która współpracuje z danym stanem. |
| String Name | Zwraca nazwę aktualnego wyświetlanego komponentu. |
| String ButtonText | Pozwala ustawić lub zwrócić nazwę przycisku w zależności od aktualnego stanu. |
| String NextState | Pozwala ustalić lub zwrócić następny stan, gdy powróciliśmy do poprzedniego. |
| String PrevState | Pozwala ustalić lub zwrócić poprzedni stan. |
| Bool AllowChangeState | Reprezentuje możliwość przejścia do następnego stanu. |
| String Title | Reprezentuje tytuł danego komponentu. |
| Bool Rebuidable | Ustala, czy stan powinien być budowany na nowo przy każdej zmianie |
| Bool SaveEnable | Ustala, czy można zapisać dane ze zmienionego stanu |
| Dictionary<string, string> DefaultNextStateData | Reprezentuje domyślne dane przekazywane do stanu następnego |
| Dictionary<string, string>DefaultPrevStateData | Reprezentuje domyślne dane przekazywane do stanu poprzedniego |
| StateChangeRequestEventargs EventData | Ustala (nadpisuje domyślne) dane przekazywane do kolejnego stanu |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Virtual void DatabaseChanged | Metoda, którą dziedziczyć będą kolejne komponenty potrzebujące implementacji odpowiedzi na zmianę bazy danych. |
| Virtual void OnPrevState | Metoda, którą implementować będą kolejne komponenty chcące wykonać funkcjonalność przed zmianą na stan poprzedni. |
| Virtual void OnNextState | Metoda, którą implementować będą kolejne komponenty chcące wykonać funkcjonalność przed zmianą na stan następny. |
| Virtual void ModifyAllowChangeState | Metoda, która dziedziczyć będą kolejne komponent potrzebujące implementacji zmiany możliwości właściwości AllowChangeState. |

|  |  |
| --- | --- |
| **Klasa** | StateControl |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Virtual StateChangeRequestEventargs EventData | Pozwala dziedziczyć przyszłym kontrolkom tę właściwość by mogły otrzymywać konieczne dane do operacji. |
| Virtual void OnResize | Pozwala dziedziczyć przyszłym kontrolkom tę właściwość by mogły ustalać szerokość i wysokość danego komponentu. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

***6.3.2 Stany widoku***

***6.3.2.1 Partials***

|  |  |
| --- | --- |
| **Klasa** | DatabasesListView |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Static string DatabaseNameToAction | Pozwala pobrać aktualną bazę danych |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Event EventHandler DatabaseAdd | *Informuje obserwatorów o dodaniu nowej bazy danych.* |
| Event EventHandler DatabaseDelete | *Informuje obserwatorów o usunięciu zaznaczonej bazy danych.* |
| Event EventHandler CurrentlyUsedDatabaseDelete | *Informuje obserwatorów o usunięciu bazy danych, która jest aktualnie w użyciu.* |
| Void Init | *Metoda ta pozwala wyświetlić listę baz danych.* |
| Void DatabaseAddHandler | *Obsługuje zdarzenie dodania nowej bazy danych.* |
| Void DatabaseDeleteHandler | *Obsługuje zdarzenie usunięcia bazy danych.* |
| Void CurrentlyUsedDatabaseDeleteHandler | *Obsługuje zdarzenie usunięcia aktualnie używanej bazy danych.* |

***6.4.2.1 RecordsListing***

|  |  |
| --- | --- |
| **Klasa** | RecordsListing |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

|  |  |
| --- | --- |
| **Klasa** | RecordsListingModel |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| List<string> TableFieldNames | Pozwala ustalić i zwrócić listę nazw pól tabeli. |
| List<List<string>> RecordsData | Pozwala ustalić i zwrócić wartości rekordów w poszczególnych kolumnach. |
| Int CurrentPage | Pozwala ustalić oraz zwrócić aktualna stronę rekordów. |
| String SelectedRecordID | Pozwala ustalić oraz zwrócić ID aktualnie wybranego rekordu. |
| Int AmountOfPages | Pozwala ustalić oraz zwrócić ilość stron potrzebnych na rekordy. |
| String TableName | Pozwala ustawić i zwrócić aktualną nazwę tabeli. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| - | - |

|  |  |
| --- | --- |
| **Klasa** | RecordsListingPresenter |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| List<string> TableFieldNames | Pozwala ustalić i zwrócić listę nazw tabel w modelu. |
| List<List<string>> RecordsData | Pozwala ustalić i zwrócić wartości rekordów w poszczególnych kolumnach w modelu. |
| Int CurrentPage | Pozwala ustalić oraz zwrócić aktualna stronę rekordów w modelu. |
| String SelectedRecordID | Pozwala ustalić oraz zwrócić ID aktualnie wybranego rekordu w modelu. |
| Int AmountOfPages | Pozwala ustalić oraz zwrócić ilość stron potrzebnych na rekordy w modelu. |
| String TableName | Pozwala ustawić i zwrócić aktualną nazwę tabeli w modelu. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Void Init | *Ustawia początkowe wartości właściwości.* |
| Void UpdateData | *Aktualizuje dane kontrolki i je wyświetla.* |
| Void DeleteRecord | *Usuwa rekord z bazy danych, aktualizuje dane i odświeża kontrolkę.* |
| StateChangeRequestEventArgs GetSelectedRecordArgs | *Pozwala zwrócić dane aktualnie zaznaczonego rekordu.* |
| Dictionary<string, string> GetSelectedRecordData | *Zwraca zbiór nazwa kolumny – wartość.* |

|  |  |
| --- | --- |
| **Klasa** | RecordsListingView |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Override StateChangeRequestEventArgs EventData | Pozwala ustalić dane wysłane przez wcześniejszy stan. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Action<string> SetTitle | *Pozwala przypiąć metodę, która odpowiada za zmianę tytułu kontrolki.* |
| Override void OnResize | *Pozwala zaktualizować rozmiary kontrolek.* |
| Void UpdateView | *Aktualizuje wygląd poszczególnych kontrolek w widoku.* |
| Public void ToggleEditButtonVisibility | *W zależności od tego, czy zaznaczony jest rekord, pokazuje lub ukrywa przyciski Edycji i Usunięcia rekordu.* |

***6.4.2.1.1 Partials***

|  |  |
| --- | --- |
| **Klasa** | Record |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String ID | Pozwala pobrać lub ustalić ID rekordu. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Event EventHandler SelectedRecordChange | Informuje obserwatorów o zdarzeniu zmiany zaznaczenia rekordu. |

|  |  |
| --- | --- |
| **Klasa** | RecordField |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String FieldValue | Pozwala pobrać lub ustalić nazwę pola. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

|  |  |
| --- | --- |
| **Klasa** | TableHeader |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

|  |  |
| --- | --- |
| **Klasa** | TableHeaderItem |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String FieldName | Pozwala ustalić nazwę pola tabeli w kontrolce. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

***6.4.2.2 RowEditor***

|  |  |
| --- | --- |
| **Klasa** | RowEditor |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Void SetTitle | *Pozwala ustalić tytuł kontrolki w zależności od argumentów (edycja lub dodawanie)* |
| Override void OnNextState | *Wykonuje zapisanie danych z kontrolki.* |

|  |  |
| --- | --- |
| **Klasa** | RowEditorControl |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Bool ReturnChangeRowControl | *Zwraca czy stan jest aktualnie edycją, czy dodawaniem nowego rekordu.* |
| String TableName | *Pozwala ustalić lub zwrócić nazwę tabeli.* |
| Override StateChangeRequestEventArgs EventData | Pozwala ustalić początkowe wartości w zależności od otrzymanych danych. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Action<string, bool> SetTitle | *Pozwala podpiąć metodę zmieniającą tytuł kontrolki w zależności od parametrów.* |
| Void Save | *W zależności od stanu kontrolki, próbuje edytować lub zapisać nowy rekord do bazy danych. W przypadku błędu pojawia się komunikat.* |

***6.4.2.2.1 Partials***

|  |  |
| --- | --- |
| **Klasa** | FieldEditor |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String FieldName | Pozwala ustalić lub zwrócić tekst z kontrolki nazwy kolumny rekordu. |
| String Value | Pozwala ustalić lub zwrócić tekst z kontrolki wartości rekordu. |
| String FieldType | Pozwala ustalić lub zwrócić tekst z kontrolki typu danych rekordu. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | **-** |

|  |  |
| --- | --- |
| **Klasa** | HorizontalLineControl |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String FieldName | Jest to odziedziczona właściwość po interfejsie bez implementacji. |
| String ValueName | Jest to odziedziczona właściwość po interfejsie bez implementacji. |
| Bool NullValue | Jest to odziedziczona właściwość po interfejsie bez implementacji. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Bool IfNecesseryFullfiled | Jest to odziedziczona metoda po interfejsie bez implementacji. |

|  |  |
| --- | --- |
| **Interfejs** | IControlInterface |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String FieldName | Brak implementacji |
| String ValueName | Brak implementacji |
| Bool NullValue | Brak implementacji |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Bool IfNecesseryFullfiled | Brak implementacji |

|  |  |
| --- | --- |
| **Klasa** | OneColumnDropDownEditor |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String FieldName | Pozwala ustalić wartość w kontrolce odpowiadającej za nazwę kolumny. Jeżeli ma zwrócić nazwę, obcina ewentualną gwiazdkę z końca nazwy oznaczającej konieczność wpisania wartości do kontrolki. |
| String ValueName | Zwraca lub ustala wartość w kontrolce odpowiadającej za wartość. |
| Bool NullValue | Pozwala ustalić czy w kontrolce została wybrana pusta wartość. |
| String Type | Pozwala ustalić typ danych. |
| String[] Values | Pozwala ustalić lub pobrać listę wartości z kontrolki. |
| Metody publiczne | |
| Void SelectItem | *Z listy wybiera pierwszą wartość.* |
| Void SelectItem | *Za pomocą argumentu wybiera wartość w liście.* |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Bool IfNecesseryFullfiled | Sprawdza, czy wszystkie wartości w kontrolce, które są wymagane są wypełnione. Zwraca prawdę jeżeli tak, w przeciwnym wypadku zwraca fałsz i ustala błąd. |

|  |  |
| --- | --- |
| **Klasa** | OneColumnTextEditor |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String FieldName | Pozwala ustalić wartość w kontrolce odpowiadającej za nazwę kolumny. Jeżeli ma zwrócić nazwę, obcina ewentualną gwiazdkę z końca nazwy oznaczającej konieczność wpisania wartości do kontrolki. |
| String ValueName | Zwraca lub ustala wartość w kontrolce odpowiadającej za wartość. |
| Bool NullValue | Pozwala ustalić czy w kontrolce została wybrana pusta wartość. |
| String Type | Pozwala ustalić typ danych lub pobrać typ danych. |
| String Length | Pozwala ustalić lub pobrać maksymalną długość wpisywanego tekstu do kontrolki dla wartości. |
| Metody publiczne | |
| Void SelectItem | *Z listy wybiera pierwszą wartość.* |
| Void SelectItem | *Za pomocą argumentu wybiera wartość w liście.* |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Bool IfNecesseryFullfiled | Sprawdza, czy wszystkie wartości w kontrolce, które są wymagane są wypełnione. Zwraca prawdę jeżeli tak, w przeciwnym wypadku zwraca fałsz i ustala błąd. |

***6.4.2.3 TableEditor***

|  |  |
| --- | --- |
| **Klasa** | TableEditor |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Void SetTitle | *Ustawia tytuł stanu (wyświetlany w widoku)* |
| Override void ModifyAllowChangeState | *Najpierw kontrolka musi przejść przez wszystkie testy, które pozwalają stworzyć lub zedytować poszczególną tabelę (jak nazwa tabeli, ilość kolumn) lub kolumnę. Następnie:*  *W przypadku kreatora:*  *Przy kreatorze tworzy nowa tabele i ustala AllowStateChange na true.*  *W przypadku edytora:*  *Sprawdza najpierw, czy nowa nazwa tabeli różni sięod starej. Jeżeli tak, zmienia ją na nową.*  *Następnie sprawdzając po kolei wszystkie kolumny starej tabeli z nową zmienia je jeżeli są różne pod względem właściwości.*  *Następnie jeżeli nowa tabela ma większą ilość kolumn, dodaje do tej tabeli te kolumny. Na końcu ustala AllowStateChange na true.*  *Jeżeli testy nie przejdą, AllowStateChange jest ustalany na false i nie jest możliwe zapisanie zmian.* |
| Void SetTitle | *Pozwala ustalić tytuł kontrolki.* |

|  |  |
| --- | --- |
| **Klasa** | TableEditorControl |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Override StateChangeRequestEventArgs EventData | Gdy ustalono tryb edytora wyświetla wszystkie kolumny. Gdy ustalono tryb kreatora, wyświetla pojedynczą pustą kontrolkę dla kolumny. |
| Mode ActualMode | Pozwala ustalić lub zwrócić jaki aktualny stan jest przy tworzeniu (edycja czy tworzenie nowej tabeli). |
| String OldTableName | Pozwala ustalić lub zwrócić nazwę tabeli przed zmianą. |
| Override StateChangeRequestEventArgs | Poprzez setter pozwala wykonać wszystkie konieczne operacje do stworzenia możliwości stworzenia nowej tabeli lub wylistowania wszystkich koniecznych właściwości już istniejącej tabeli. |
| String NewTableName | Zwraca lub ustawia tekst w kontrolce odpowiadającej za nazwę tabeli. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Action<string> SetTitle | *Pozwala podpiąć metodę, która zmieni tytuł kontrolki.* |
| List<ColumnStructureCreator> GetColumnsWIthProperties | *Tworzy i zwraca listę ColumnStructureCreator’ow na podstawie kontrolek w TableEditorze.* |
| Bool CheckTableName | *W przypadku, gdy nowa nazwa tabeli jest pusta, zwraca fałsz.* |
| Bool CheckAmountOfColumns | *W przypadku, gdy kontrolka zawiera mniej niż jedną kolumnę zwraca fałsz.* |
| Bool CheckNamesOfColumns | *Sprawdza czy wszystkie kontrolki reprezentujące kolumnę zawierają jakąś nazwę. Jeżeli tak, zwraca prawdę.* |
| Bool CheckTypesOfColumns | *Sprawdza czy wszystkie kontrolki reprezentujące kolumnę mają wybrany typ. Jeżeli tak jest, to zwraca prawdę.* |
| Int GetAmountOfPrimaryKeys | *Zwraca ilość kontrolek, w których wybrany jest klucz główny.* |
| List<string> GetAllFieldNamesWherePrimary | *Zwraca listę nazw pól, które są zaznaczone jako klucze główne.* |
| List<ColumnStructureCreator> GetAllColumns | *Na podstawie kontrolek w zbiorniku tworzy wewnętrzną listę obiektów typu ColumnStructureCreator.* |
| Override void OnResize | *Aktualizuje rozmiary kontrolki.* |

***6.4.2.3.1 Partials***

|  |  |
| --- | --- |
| **Klasa** | FieldEditor |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| ColumnStructureCreator ColumnStructureObject | Na podstawie właściwości zwraca obiekt ColumnStructureCreator. |
| String FieldName | Pozwala ustalić tekst w kontrolce odpowiadającej za nazwę pola. |
| String FieldType | Pozwala zwrócić lub ustalić typ w kontrolce. |
| String TypeLength | Pozwala ustalić lub zwrócić długość typu z kontrolki. |
| Bool NullValue | Pozwala ustalić lub zwrócić czy z kontrolki czy kolumna może zawierać puste wartości. |
| Bool Primary\_Key | Pozwala ustalić lub zwrócić czy z kontrolki czy kolumna jest kluczem głównym. |
| String Default | Pozwala ustalić lub zwrócić domyślny tekst jaki będzie wpisywany jako wartość dla kolumny. |
| Bool Extra | Pozwala ustalić lub zwrócić czy z kontrolki czy kolumna jest typu auto\_increment. |
| String TableNameItBelongs | Pozwala ustalić lub zwrócić nazwę tabeli do której dana kolumna należy. |
| Bool EditorTableMode | Pozwala ustalić lub zwrócić, czy aktualnym trybem jest edycja czy dodawanie nowej tabeli. |
| Bool LengthReadOnly | Pozwala ustalić lub zwrócić to, czy kontrolka długość typu danych jest tylko do odczytu. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Void Clear | *Czyści kontrolki do stanu początkowego.* |
| Void Auto\_Increment\_CheckChanged | *Metoda odpowiadająca na zdarzenie zaznaczenia pola auto\_increment. Ustala ona logikę pola z właściwością auto\_increment.* |
| Void Primary\_Key\_CheckChanged | *Metoda odpowiadająca na zdarzenie zaznaczenia pola Primary Key. Ustala ona logikę pola z właściwością Primary Key.* |
| Void Foreign\_Key\_Check\_Changed | *Metoda odpowiadająca na zdarzenie zaznaczenia pola Foreign Key. Ustala ona logikę pola z właściwością Foreign Key.* |
| Void Field\_Type\_Changed | *Metoda odpowiadająca na zdarzenie wybrania typu pola. W przypadku wybrania odpowiedniego pola umożliwienie jest ustalenie jego dodatkowej właściwości.W przypadku wyboru odpowiedniego typu danych uruchamiane są dodatkowe mechanizmy odpowiedzialne za pomoc użytkownika w doborze długości danych.* |

|  |  |
| --- | --- |
| **Klasa** | FieldEditorPresenter |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Event EventHandler PrimaryKeyCheckChanged | Informuje obserwatorów o wystąpieniu zdarzenia zaznaczenia pola Primary Key. |
| Event EventHandler ForeignKeyCheckChanged | Informuje obserwatorów o wystąpieniu zdarzenia zaznaczenia pola Foreign Key. |
| Event EventHandler AutoIncrementCheckChanged | Informuje obserwatorów o wystąpieniu zdarzenia zaznaczenia pola Auto Increment. |
| Event EventHandler FieldTypeChanged | Informuje obserwatorów o wystąpieniu zdarzenia wybrania typu pola. |

***6.4.2.4 TablesListing***

|  |  |
| --- | --- |
| **Klasa** | TablesListingControl |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| - | - |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Void AddTables | *Czyści listę aktualnych tabel i dodaje nową listę na podstawie aktualnej bazy danych.* |
| Void OnResize | *Zmienia szerokość kontrolki związanej z nazwą tabeli.* |

|  |  |
| --- | --- |
| **Klasa** | TablesListing |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

***6.4.2.5.1 Partials***

|  |  |
| --- | --- |
| **Klasa** | TableItem |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

***6.4.2.6 Welcome***

|  |  |
| --- | --- |
| **Klasa** | Welcome |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

|  |  |
| --- | --- |
| **Klasa** | WelcomeControl |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

***6.4.2.6 Klasy reprezentujące główny widok***

|  |  |
| --- | --- |
| **Klasa** | View |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Event Action<int,int> ContainerResize | *Pozwala podpiąć event zmiany rozmiarów kontenera.* |
| Void DisplayStateChange | *Ustala rozmiary poszczególnych kontrolek, podstawowe ustawienia i aktualizuje widok.* |
| Void DisplayError | *Wyświetla okienko z informacją błędu.* |

|  |  |
| --- | --- |
| **Klasa** | Presenter |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| State ActiveState | Zwraca komponent, który jest aktualnie wyświetlany w aplikacji. |
| Action<string> StateBuilt | Zdarzenie wysyłane w momencie utworzenia / przebudowy stanu. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Void Init | *Początkowe ustawienia, podpinanie się pod eventy.* |
| Void ChangeState | *Zmienia skupienie aplikacji na inny stan i aktualizuje widok.* |
| State GetStateByName | *Zwraca obiekt stanu przy pomocy nazwy komponentu.* |

|  |  |
| --- | --- |
| **Klasa** | Model |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| StatesDict States | Pozwala pobrać listę stanów aplikacji. |
| State ActiveState | Pozwala ustalić lub zwrócić aktualny stan aplikacji. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Event Action<string> StateBuilt | *Wysyła zdarzenie po utworzeniu / przebudowaniu stanu.* |
| Void RebuildState | *Buduje stan na nowo.* |
| Void CreateStates | *Tworzy początkowe stany, jakie będzie przyjmować aplikacja.* |

**7\* Testy**

Program przetestowano na bazie świat jak i innych bazach (wbudowanych w MySQL jak specjalnych bazach testowych). Testowanie polegało na dodawaniu / usuwaniu baz, dodawaniu / usuwaniu / edycji tabel jak i rekordów.

**8. Instrukcja użytkownika**

Przed rozpoczęciem pracy w programie DB Editor konieczne jest uruchomienie serwera SQL.

Dwukrotnym kliknięciem lewym przyciskiem myszy otwieramy plik o nazwie db-editor.exe. Pojawi się okienko do logowania, w którym wpisujemy naszą nazwę użytkownika oraz hasło. Jeżeli serwer SQL jest uruchomiony na naszym komputerze, to w polu Server pozostawiamy „localhost”, jeżeli chcemy się podłączyć do zewnętrznego serwera SQL, to w miejscu Server podajemy IP tego serwera. Następnie klikamy przycisk „Connect”.

Przy poprawnym połączeniu się z bazą danych pojawi się główne okno do pracy.

Po lewej stronie aplikacji znajduje się lista baz danych, którą obecnie może obsługiwać użytkownik, a niżej jest możliwość dodania nowej bazy danych.

Jeżeli chcemy usunąć bazę danych, wystarczy zaznaczyć kliknięciem myszy daną bazę i nacisnąć wyżej czerwony przycisk Delete. Po potwierdzeniu baza zostanie usunięta, a lista baz odswieżona.

Wybór bazy danych, na której chcemy pracować wykonuje się poprzez dwukrotne kliknięcie myszą w nazwę bazy danych.

W dowolnej chwili bazę danych można zmienić klikając dwukrotnie na inna nazwę bazy danych.

Po wybraniu bazy danych, po prawej stronie pojawi się lista tabel w danej bazie danych wraz z opcjami edycji. W dowolnej chwili można dodać nową tabelę do bazy danych klikając w dolnym prawym rogu aplikacji „Add new Table”

Dodawanie nowej tabeli

Po kliknięciu w ten niebieski przycisk, pojawi się nam okno do tworzenia tabeli. W górnej części znajduje się pole tekstowe, w którym musimy podać nazwę tabeli (jest to pole wymagane).

Niżej już pojawiła się pojedyncza kontrolka odpowiadająca za kolumnę w tabeli. Kolejne właściwości kontrolki są podpisane, wystarczy uzupełnić kontrolkę w pożądany przez siebie sposób oraz taki, w jaki pozwala nam na to kontrolka (kontrolka pilnuje większość zachowań niedozwolonych przy tworzeniu nowej kolumny).

Aby dodać nową kolumnę do tej bazy danych, wystarczy nacisnąć przycisk „Add new field” poniżej i ponownie uzupełnić daną kontrolkę. Jeżeli jednak zdecydujemy o tym, że nie chcemy mieć danej kolumny w tabeli, wystarczy nacisnąć przycisk Delete. W przypadku, gdy edytujemy już istniejące tabele, powstanie zapytanie, czy na pewno chcemy usunąć daną kolumnę.

Gdy skończyliśmy edytować naszą nową kolumnę, wystarczy nacisnąć przycisk „Save”. Jeżeli widok powróci do listy tabel (wraz z naszą nowo utworzoną tabelą), oznacza to, że operacja się powiodła. W przeciwnym razie na ekranie pojawi się znak wykrzyknika, który po najechaniu myszką wskaże jaki problem należy rozwiązać, aby móc zapisać daną tabelę.

Dane konieczne, by utworzyć nową tabelę to:

-nazwa tabeli,

-przynajmniej jedna kolumna,

-kolumna ta musi posiadać nazwę oraz wybrany typ danych.

Przy wyświetlaniu listy tabel mamy również inne możliwości, które zostały podzielone na trzy części:

Edycja tabel (Edit)

Kliknięcie w ten przycisk wyświetli nam widok podobny do dodawania nowej tabeli, z tą różnicą, że kontrolka będzie wypełniona już komponentami zgodnymi z tabelą. W tym momencie możemy znowu dowolnie zmieniać wartości w naszych kontrolkach (jednak należy robić to z uwagą, gdyż ewentualne błędnie zaimplementowane właściwości mogą doprowadzić do utraty danych).

Aby zmienić nazwę tabeli, wystarczy wpisać nową nazwę w odpowiednim miejscu.

Również możemy dodawać nowe kolumny i je wypełniać. Żeby zmiany weszły w życie należy nacisnąć przycisk „Save”.

Wyświetlenie rekordów (Records)

Kliknięcie w ten przycisk spowoduje wyświetlenie pierwszych pięćdziesięciu rekordów (o ile istnieją) z wybranej tabeli. Aby przeglądać kolejne strony (strona zawiera w sobie 50 kolejnych rekordów) rekordów wystarczy nacisnąć przycisk „Next Page”, a aby powrócić do poprzedniej „Prev Page”

Zostaną nam również udostępnione nowe możliwości:

Edytowanie rekordu (Edit)

Edycję konkretnego rekordu rozpoczynamy od zaznaczenia go w poprzednim stadium. Następnie naciskamy przycisk Edit i następnie pojawi się lista kontrolek z nazwą, jego typem oraz wartością, jaką ma wpisaną.

Wystarczy zmienić wartości na takie, które są zgodne z typem danych, a następnie kliknąć przycisk Save.

Usuwanie rekordu (Delete)

Usuwanie rekordu rozpoczynamy znów od zaznaczenia go w poprzednim stanie. Po kliknięciu w przycisk Delete pojawi się pytanie czy na pewno chcemy tego dokonać. Po potwierdzeniu rekord zostanie usunięty, a lista rekordów odswieżona.

Dodawanie nowego rekordu (Add new record)

Po kliknięciu w przycisk pojawi się listą kontrolek. Każda kontrolka składa się z 3 rzeczy.

1. Nazwa kolumny
2. Typ danych wprowadzanych
3. Miejsce do wpisania wartości/wybrania wartości z rozwijanej listy

Jeżeli w nazwie kolumny wartości pojawia się gwiazdka (\*), oznacza to, że kolumna ta musi być wypełniona. Brak takiej gwiazdki oznacza możliwość pozostawienia tego pola niewypełnionego.

Aplikacja pilnuje, by nie przekraczać maksymalnej długości wpisywanej wartości dla danej kolumny.

Gdy zakończymy dodawanie nowego rekordu według wymagań, klikamy „Save”.

Usunięcie tabeli z bazy danych (Delete)

Kliknięcie w ten przycisk spowoduje wyświetlenie się pytania, czy na pewno chcemy usunąć tę tabelę z bazy danych. Po potwierdzeniu baza zostanie usunięta.