|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Dodanie kolejnych informacji z klas, dodanie kolejnej części instrukcji użytkownika | D | 25.06.2016 | Kamil Kryus |  |

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 **CZYM?**

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. **COŚ O TYM, ŻE MOŻNA PRZEJSC DO BEZPOSREDNIEJ STRONY Z REKORDAMI.**

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

**Edycja polega na:**

W każdej chwili jesteśmy w stanie dodać nowy rekord poprzez przycisk „Add new record” w dolnym prawym rogu programu.

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,

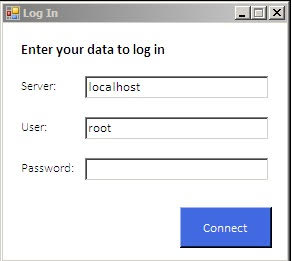
-zmiana poszczególnych lub wszystkich wartości w rekordzie.

2.2 Wymagania niefunkcyjne

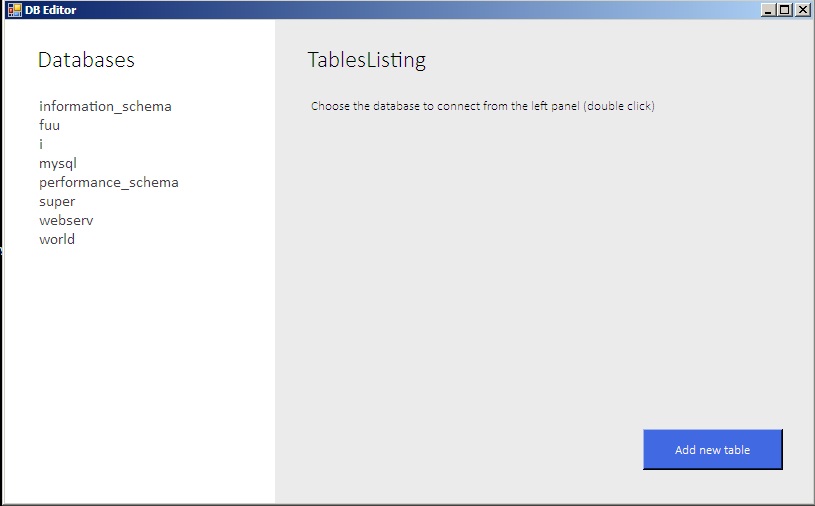
**3 Projekt interfejsu graficznego**

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

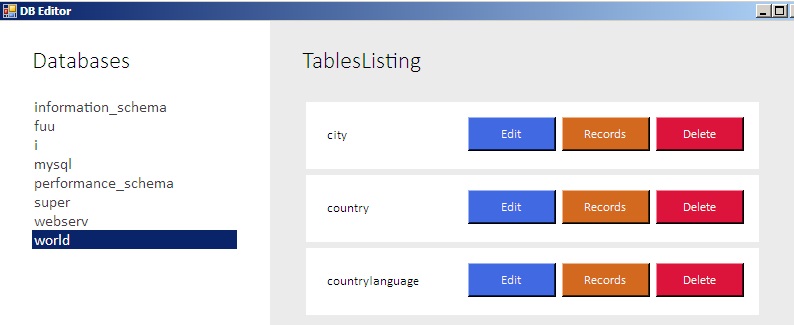
*Główne okno do logowania:*

**

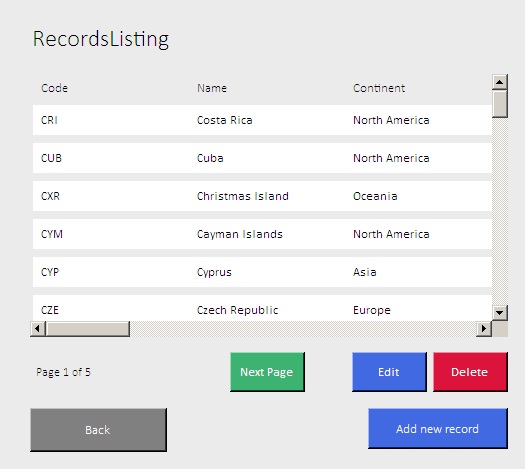
*Główne okno po zalogowaniu się użytkownika:*

**

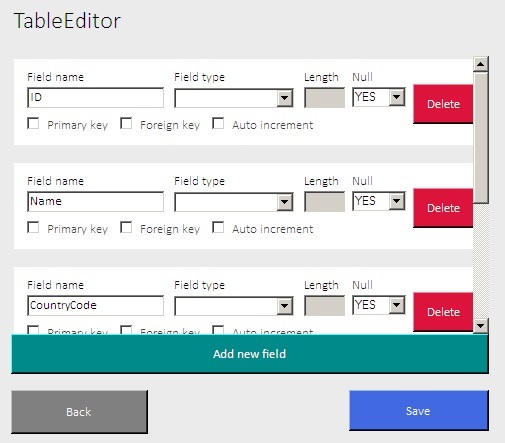
*Lista tabel po wybraniu bazy danych:*

**

*Lista rekordów po wybraniu tabeli:*

**

*Możliwość edycji pół:*

**

**4. Model obiektowy**

*Schemat UML klas*

**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 bazę danych o podanej nazwie. Zwraca strukturę OperationResult.* |
| OperationResult CreateTable | *Tworzy tabelę o podanej nazwie tworząc przy tym kolumny zadane w argumencie oraz przypisuje klucze główne i obce. 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.* |
| 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.* |
| 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 oraz ich właściwości z danej tabeli. Zwraca listęw kolejnych właściwości pól.* |
| List<ColumnStructureCreator> GetColumnSctuctureCreatorsFromTable | *Wykonuje zapytanie do bazy danych i na podstawie zwróconego wyniku tworzy listę kolumn przedstawianych jako obiekty ColumnStructureCreator.* |

|  |  |
| --- | --- |
| **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.* |
| List<List<string>> GetFirstPageOfRecords | *Wykonuje zapytanie do bazy danych o pierwsze pięćdziesiąt rekordów danej tabeli. Zwraca listę, która zawiera listę poszczególnych pól w rekordzie.* |
| 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 nazw pól w rekordzie.* |

|  |  |
| --- | --- |
| **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.* |

|  |  |
| --- | --- |
| **Klasa** | ColumnStructureCreator |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| string Field | *Reprezentuje nazwę kolumny.* |
| string Type | *Reprezentuje typ kolumny.* |
| 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.* |
| 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.* |
| Object this | *Operator indeksowania, który przy kolejnym indeksie zwraca kolejną właściwość. Jeżeli indeks wyjdzie poza zakres, jest automatycznie skracany do możliwej wartości indeksu.* |

|  |  |
| --- | --- |
| **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.** |
| 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 | ***Otwiera i zamyka połączenie z bazą danych.*** |

***6.2 Klasa reprezentująca informacje o danym produkcie***

|  |  |
| --- | --- |
| **Klasa** | AboutBox |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String AssemblyTitle | **DO DOKONCZENIA** |
| String AssemblyVersion | **DO DOKONCZENIA** |
| String AssemblyDescription | **DO DOKONCZENIA** |
| String AssemblyProduct | **DO DOKONCZENIA** |
| String Copyright | **DO DOKONCZENIA** |
| String AssemblyCompany | **DO DOKONCZENIA** |

***6.3 Klasy odpowiedzialne za uwierzytelnianie***

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

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

|  |  |
| --- | --- |
| **Klasa** | Model[LoginWindow] |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
|  |  |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |

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

***6.4.1 Definicje (LUB NAZWIJ JAKOS INACZEJ TO)***

**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 komponentu, na którym jest „skupienie”.** |
| 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.** |
| StateChangeRequestEventargs EventData | **TEGO NIE WIEM** |
| Bool AllowChangeState | Reprezentuje możliwość przejścia do następnego stanu. |
| String Title | **NIE WIEM** |
| Bool Rebuidable | **NIE WIEM** |
| 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 dziedziczyć będą kolejne komponenty chcące wrócić do stanu poprzedniego.** |
| Virtual void OnNextState | **Metoda, którą dziedziczyć będą kolejne komponenty chcące wrócić do następnego stanu z poprzedniego.** |
| 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 tą właściwość by… DO DOKOŃCZENIA** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

***6.4.2 Stany widoku (LUB NAZWIJ TO INACZEJ JAKOS)***

***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 tabel.** |
| 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.*** |

|  |  |
| --- | --- |
| **Klasa** | RecordsListingView |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Override StateChangeRequestEventArgs EventData | **Pozwala ustalić COŚ.** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Void UpdateView | ***Aktualizuje wygląd poszczególnych kontrolek w widoku.*** |

***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 tekst 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* |
| **-** | ***-*** |

|  |  |
| --- | --- |
| **Klasa** | RowEditorControl |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Override StateChangeRequestEventArgs EventData | **Pozwala ustalić … COŚ** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| **-** | ***-*** |

***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. |
| ColumnStructureCreator ColumnStructureObject | Pozwala zwrócić obiekt ColumnStructureCreator na podstawie właściwości w kontrolce. |
| String FieldType | Ustawia wartość w kontrolce. Zwraca wartość z kontrolki jeżeli jest jakaś wybrana, jeżeli nie, to zwraca pustego stringa. |
| String TypeLength | Ustawia lub zwraca tekst z kontrolki związanej z długością typu danych. |
| Bool NullValue | Jeżeli w kontrolce zostałą wybrana wartość “YES” to obiekt zwraca prawdę, jeżeli przypisana zostaje wartość true, kontrolka zaznacza wartość “YES”. |
| Bool Primary\_Key | Ustawia lub zwraca wartość zaznaczenia checkboxa odpowiedzialnego za Primary Key. |
| String Default | Ustawia lub zwraca tekst z kontrolki Default. |
| Bool Extra | Ustawia lub zwraca wartość zaznaczenia kontrolki auto\_increment. |
| String TableNameItBelongs | Ustawia lub zwraca nazwę tabeli na której aktualnie pracujemy. |
| Bool EditorTableMode | Zwraca wartość czy aktualny stan pracy na aplikacji jest typem edycji. |
| Bool LengthReadOnly | Ustawia lub zwraca własność ReadOnly kontrolki długości typu danych. |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Void Clear | **Czyści kontrolkę.** |
| Void Auto\_Increment\_CheckChanged | Ustawia odpowiednio kontrolkę przy kliknięciu w checkboxa odpowiadającego za auto\_increment. |
| Void Primary\_Key\_CheckChanged | Ustawia odpowiednio kontrolkę przy kliknięciu w checkboxa odpowiadającego za primary\_key |
| Void ForeignKey\_CheckChanged | Ustawia odpowiednio kontrolkę przy kliknięciu w checkboxa odpowiadającego za foreign\_key |
| Void Field\_Type\_Changed | Ustawia odpowiednio kontrolkę przy kliknięciu w checkboxa odpowiadającego za wybor typu danych. |

***6.4.2.3 TableEditor***

|  |  |
| --- | --- |
| **Klasa** | TableEditor |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Override void OnNextState | ***TUTAJ BĘDZIE JAKAS ‘INKSZA’ FUNKCJONALNOSC*** |
| Override void OnPrevState | ***TUTAJ BĘDZIE JAKAS ‘INKSZA’ FUNKCJONALNOSC*** |
| Void SetTitle | ***TUTAJ BĘDZIE JAKAS FUNKCJONALNOSC*** |
| 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.* |

|  |  |
| --- | --- |
| **Klasa** | TableEditorControl |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| Override StateChangeRequestEventArgs EventData | **Pozwala ustalić COŚ.** |
| 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 | ***COŚ ROBI BARDZO CIEKAWEGO*** |
| 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.* |
| Bool CheckAmountOfPrimaryKeys | *Jeżeli w zbiorniku jest więcej, aniżeli jeden klucz główny zwraca fałsz.* |
| List<ColumnStructureCreator> GetAllColumns | *Na podstawie kontrolek w zbiorniku tworzy wewnętrzną listę obiektów typu ColumnStructureCreator.* |

***6.4.2.3.1 Partials***

|  |  |
| --- | --- |
| **Klasa** | FieldEditor |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| String FieldName | **Pozwala ustalić tekst w kontrolce odpowiadającej za nazwę pola.** |
| 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.* ***MOŻLIWE COŚ JESZCZE, JEŻELI BĘDZIE SIĘ WYBIERAŁO TEN KLUCZ OBCY Z INNEJ TABELI*** |
| Void Field\_Type\_Changed | *Metoda odpowiadająca na zdarzenie wybrania typu pola. W przypadku wybrania odpowiedniego pola umożliwnienie jest ustalenie jego dodatkowej właściwości.* |

|  |  |
| --- | --- |
| **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 TableOverview***

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

***6.4.2.5 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* |
| Override void DatabaseChanged | *Pobiera listę baz danych oraz wyświetla je.* |

***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 Klasy reprezentujące główny widok***

|  |  |
| --- | --- |
| **Klasa** | View |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| **-** | **-** |
| Metody publiczne | |
| typ NazwaMetody | *Jak działa i co zwraca* |
| Event EventHandler DatabaseChanged | *Informuje obserwatorów o zmianie bazy danych.* |
| Void DisplayStateChange | ***Ustala rozmiary poszczególnych kontrolek, podstawowe ustawienia i aktualizuje widok.*** |

|  |  |
| --- | --- |
| **Klasa** | Presenter |
| **Właściwości** | |
| typ NazwaWlasciwosci | *Co reprezentuje i jak działa* |
| State ActiveState | **Zwraca komponent, na którym jest aktualnie skupiona aplikacja.** |
| 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.*** |
| 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* |
| **-** | ***-*** |

**7\* Testy**

**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:

**<<TUTAJ COŚ DALEJ>>**

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)

**<<TUTAJ COŚ MA BYĆ>>**

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.