Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б. С. Галущака»

**Практическая работа 5**

Выполнил: студент группы ПР–21.106  
Бушин А.Н.

2024 год

**1 Анализ предметной области**

Предметная область включает в себя разработку приложения по изучению иностранных языков с использованием базы данных Entity Framework. Ключевые компоненты:

* MainWindow класс, представляющий основное окно приложения.
* Класс helper для управления контекстом базы данных и получения экземпляра EnglishEntities.
* EnglishEntities, наследующий DbContext, для взаимодействия с базой данных и предоставления доступа к таблицам, таким как Dictonary, Language, Lections, Levels, Progress, sysdiagrams, Test и Users.

**2 Определение функций предметной области**

MainWindow служит для управления основным окном приложения и включает статическое свойство Instance для обеспечения доступа к единственному экземпляру окна.

Класс helper отвечает за управление контекстом базы данных и предоставление доступа к экземпляру EnglishEntities.

EnglishEntities определяет структуру базы данных и обеспечивает доступ к различным таблицам, представляющим данные для изучения иностранного языка.

**3 Схемы документопотока**

Необходимые данные и документы включают в себя:

* Данные о пользователях (Users) - регистрационные данные и информация о прогрессе обучения.
* Учебные материалы (Dictonary, Language, Lections, Levels, Progress, Test) - данные об изучаемом языке, словарь, уровни, прогресс и тесты.
* Вспомогательные данные (sysdiagrams) - для поддержки метаданных базы данных.

**4 Выделенные сущности, атрибуты и установленные связи**

Сущности: Users, Dictonary, Language, Lections, Levels, Progress, Test.

Атрибуты: Идентификаторы, регистрационные данные, уровни владения языком, прогресс, содержание уроков, результаты тестов.

Связи: Взаимосвязи между таблицами для хранения информации о прогрессе пользователя, уровнях, тестах и учебных материалах.

**5 Концептуальная модель**

**Сущности:**

Users:

* Id\_user (int, PK) - идентификатор пользователя;
* Username (string) - имя пользователя;
* Password (string) - пароль пользователя;
* Email (string) - адрес электронной почты пользователя;
* RegistrationDate (DateTime) - дата регистрации пользователя.

Language:

* Id\_language (int, PK) - идентификатор языка;
* TitleEnglish (string) - название языка на английском;
* TitleRussian (string) - название языка на русском.

Lections:

* Id\_lections (int, PK) - идентификатор урока;
* Id\_language (int, FK) - идентификатор языка, связанный с уроком;
* Title (string) - название урока;
* Id\_levels (int, FK) - идентификатор уровня сложности урока;
* Status (bool) - статус урока (активен/неактивен).

Levels:

* Id\_levels (int, PK) - идентификатор уровня сложности;
* Title (string) - название уровня сложности.

Dictonary:

* Id\_dictonary (int, PK) - идентификатор слова в словаре;
* Text (string) - текст слова на оригинальном языке;
* Translate (string) - перевод слова на изучаемый язык;
* Id\_lections (int, FK) - идентификатор урока, к которому относится слово;
* Image (string) - изображение, связанное со словом.

Progress:

* Id\_progress (int, PK) - идентификатор прогресса;
* Id\_user (int, FK) - идентификатор пользователя, связанный с прогрессом;
* Id\_lections (int, FK) - идентификатор урока, связанный с прогрессом;
* Completed (bool) - статус завершения урока.

Test:

* Id\_test (int, PK) - идентификатор теста;
* Id\_lections (int, FK) - идентификатор урока, связанный с тестом;
* Questions (string) - вопросы теста;
* CorrectAnswers (string) - правильные ответы на вопросы.

Связи:

* Users <-> Progress: один пользователь может иметь несколько записей прогресса;
* Users <-> Lections: многие пользователи могут учиться на одном уроке, один пользователь может учиться на нескольких уроках;
* Language <-> Lections: один язык может иметь несколько уроков;
* Levels <-> Lections: один уровень сложности может быть присвоен нескольким урокам;
* Lections <-> Dictonary: один урок может содержать несколько слов из словаря;
* Users <-> Test: многие пользователи могут проходить один и тот же тест, один пользователь может проходить несколько тестов/

**6 Описание выходных и входных данных**

Входные данные: Регистрационные данные пользователя, данные об изучаемом языке, словарь, уровни владения языком, прогресс обучения, результаты тестов.

Выходные данные: Прогресс обучения, результаты тестов, уровни владения языком, учебные материалы, вспомогательные данные.

Вывод: Код C# предоставляет удобные средства для разработки приложения по изучению иностранных языков с использованием базы данных Entity Framework. Структура базы данных и функции приложения хорошо организованы для обеспечения эффективного управления учебными материалами и прогрессом обучения.