**2 Проектирование программного продукта**

**2.1 Постановка задачи**

В процессе выполнения дипломного проекта необходимо разработать приложение «Прикасаясь к вечности», представляющее информацию о самых старых городах и достопримечательностях Беларуси. Формирование структуры проекта необходимо начинать с определения структуры модулей, то есть, из каких частей будет состоять каждый модуль программного продукта, и какие данные будет содержать конкретная часть. Также необходимо продумать способы и методы хранения данных, удобную навигацию в модулях. Далее необходимо выбрать средства разработки, языки программирования и приступить к реализации.

Программный продукт должен содержать следующие части:

* модуль сервера,
* модуль клиента.

Серверный модуль должен содержать административную панель и API (интерфейс прикладного программирования) для обработки запросов клиента. Административная панель должна предоставлять возможности удобного добавления, редактирования и удаления данных. Для навигации по панели администратора необходимо предусмотреть меню. Важно учесть разграничение доступа в административную панель, чтобы только администраторы имели права на изменение данных в системе. API также должно быть защищено от незарегистрированных пользователей, что позволит минимизировать возможность утечки информации.

Структура серверного модуля приложения представлена на рисунке 2.1.

****

**Рисунок 2.1 – Структура модуля сервера**

Клиентский модуль должен позволять пользователю просматривать всю доступную информацию в удобном, структурированном виде. Необходимо четко разделять разделы клиентского приложения, чтобы пользователь мог интуитивно понять, что произойдет по нажатию выбранной кнопки.

Структура клиентского модуля приложения представлена на рисунке 2.2.



**Рисунок 2.2 – Структура модуля клиента**

Взаимодействие серверного и клиентского модулей отображено в приложении А.

Проектируемое приложение должно удовлетворять следующим требованиям:

* понятный пользователю и эргономичный интерфейс,
* рациональная структура приложения, с использованием удобной навигации,
* наличие поисковой системы.

Задачи, которые необходимо решить в процессе разработки программного продукта:

* формирование четкого определения и представления о предметной области,
* выбор языков программирования и среды разработки,
* разработка концепции пользовательского интерфейса,
* разработка веб-приложения для администрирования и обработки данных.

Таким образом, обозначены начальные действия при разработке дипломного проекта.

**2.2 Выбор инструментальных средств разработки**

Для реализации программного продукта использовались следующие инструменты:

* «Visual Studio 2022» – [интегрированная среда разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения.Это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, «Visual Studio» включает в себя компиляторы, средства авто завершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки [3],
* «Microsoft SQL Server» – [система управления реляционными базами данных (РСУБД)](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94). Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия [4],
* «Visual Studio Code» – это редактор исходного кода. Позиционируется как «лёгкий» редактор кода для [кроссплатформенной](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) разработки [веб](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и [облачных](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BB%D0%B0%D1%87%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) приложений. Включает в себя [отладчик](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D1%87%D0%B8%D0%BA), инструменты для работы с «Git», [подсветку синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0), «IntelliSense» и средства для [рефакторинга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3). Имеет широкие возможности для [кастомизации](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D1%82%D0%BE%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F): пользовательские темы, [сочетания клавиш](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%87%D0%B5%D1%82%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BB%D0%B0%D0%B2%D0%B8%D1%88) и [файлы конфигурации](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB_%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8) [5].
* SQL – это язык структурированных запросов, позволяющий хранить, манипулировать и извлекать данные из реляционных баз данных [6],
* «Xamarin.Forms» –платформа пользовательского интерфейса с открытым кодом. Она позволяет создавать собственные приложения с помощью много платформенного набора средств пользовательского интерфейса .NET, предназначенного для мобильных, планшетных и настольных устройств в «Android», «iOS» и других [7],
* «ASP.NET Core 5.0 MVC» – многофункциональная платформа для создания веб-приложений и API-интерфейсов с помощью структуры проектирования «Model-View-Controller». Согласно этой структуре, запросы пользователей направляются в контроллер, который отвечает за работу с моделью для выполнения действий пользователей и (или) получение результатов запросов. Контроллер выбирает представление для отображения пользователю со всеми необходимыми данными модели. [8],
* «Bootstrap 5» – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления с использованием «JavaScript» для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса [9],
* C# – современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в «.NET» [10],
* «ADO.NET Entity Framework Core» – это набор технологий в ADO.NET, которые поддерживают разработку программных приложений, ориентированных на данные [11].

**2.3 Проектирование базы данных**

На основании изученной предметной области и функциональной схемы системы разработана структура базы данных приложения.

В результате анализа были определены следующие сущности базы данных: «Roles», «Users», «Cities», «Attractions», «Games», «GameScores», «TestQuestions», «DataCategories», «ActionCategories», «DataActions», «DataReferences».

Отношение находится в первой нормальной форме (1НФ) тогда и только тогда, когда любое значение каждого кортежа (экземпляра) для каждого атрибута содержит только одно значение. Самое главное правило первой нормальной формы – атомарность.

Каждая таблица в 1НФ должна иметь первичный ключ. Он может состоять из одного или более столбцов (атрибутов). В последнем случае ключ называется составным.

Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме, и при этом любой его атрибут, не входящий в состав потенциального ключа, функционально полно зависит от каждого потенциального ключа. Функционально полная зависимость означает, что атрибут функционально зависит от всего составного потенциального ключа, но при этом не находится в функциональной зависимости от какой-либо из входящих в него частей.

В отношениях могут быть транзитивные зависимости. Третья нормальная форма позволяет избавиться от транзитивной зависимости. Задача третьей нормальной формы заключается в том, чтобы обеспечить максимальную целостность данных в базе данных. Целостность данных в базе данных обеспечивается уничтожением транзитивных зависимостей.

Таблицы находятся в третьей нормальной форме, когда они находится во второй нормальной форме, и в таблицах нет транзитивных зависимостей [12]. Таким образом, база данных приведена к третьей нормальной форме. Логическая схема базы данных отображена в приложении Б.

Результат проведенного проектирования базы данных для программного продукта можно представить в виде полного описания свойств полей для всех таблиц.

Таблица «Roles» содержит роли пользователей в системе. Поля, их типы и назначение представлены в таблице 2.1.

**Таблица 2.1 – Структура полей таблицы «Roles»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| RoleId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Name | Текстовый | Название роли, уникальное, обязательное поле |

Составлено автором

Таблица «Users» содержит информацию о пользователях. Поля, их типы и назначение представлены в таблице 2.2.

**Таблица 2.2 – Структура полей таблицы «Users»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| UserId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Email | Текстовый | Электронная почта, уникальное, обязательное поле |
| UserName | Текстовый | Имя пользователя, уникальное, обязательное поле |
| Password | Текстовый | Пароль пользователя, обязательное поле |
| RoleId | Числовой | Идентификатор роли пользователя, внешний ключ таблицы, обязательное поле |

Составлено автором

Таблица «Cities» содержит информацию о городах. Поля, их типы и назначение представлены в таблице 2.3.

**Таблица 2.3 – Структура полей таблицы «Cities»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| CityId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Title | Текстовый | Название, обязательное поле |
| Description | Текстовый | Описание, обязательное поле |

Составлено автором

Таблица «Attractions» содержит информацию о достопримечательностях. Поля, их типы и назначение представлены в таблице 2.4.

**Таблица 2.4 – Структура полей таблицы «Attractions»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| AttractionId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Title | Текстовый | Название, обязательное поле |
| Description | Текстовый | Описание, обязательное поле |

Составлено автором

Таблица «Games» служебная таблица, хранящая название игр. Поля, их типы и назначение представлены в таблице 2.5.

**Таблица 2.5 – Структура полей таблицы «Games»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| GameId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Name | Текстовый | Название игры, уникальное, обязательное поле |

Составлено автором

В таблице «GameScores» содержатся лучшие результаты пользователей по играм. Поля, их типы и назначение представлены в таблице 2.6.

**Таблица 2.6 – Структура полей таблицы «GameScores»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| GameScoreId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| UserId | Числовой | Идентификатор пользователя, внешний ключ, обязательное поле |
| GameId | Числовой | Идентификатор игры, внешний ключ, обязательное поле |
| Score | Текстовый | Результат игры, обязательное поле |

Составлено автором

Таблица «TestQuestions» с вопросами для игры. Поля, их типы и назначение представлены в таблице 2.7.

**Таблица 2.7 – Структура полей таблицы «TestQuestions»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| TestQuestionId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Question | Текстовый | Вопрос, обязательное поле |
| RightAnswer | Текстовый | Правильный ответ, обязательное поле |
| WrongAnswer1 | Текстовый | Первый неправильный ответ, обязательное поле |
| WrongAnswer2 | Текстовый | Второй неправильный ответ, обязательное поле |
| WrongAnswer3 | Текстовый | Третий неправильный ответ, обязательное поле |

Составлено автором

Таблица «DataCategories» служебная таблица, хранящая категории данных. Поля, их типы и назначение представлены в таблице 2.8.

**Таблица 2.8 – Структура полей таблицы «DataCategories»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| DataCategoryId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Category | Текстовый | Название категории данных, обязательное поле |

Составлено автором

Таблица «DataActions» связующая таблица, между категориями данных и категориями действий. Поля, их типы и назначение представлены в таблице 2.9.

**Таблица 2.9 – Структура полей таблицы «DataActions»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| DataActionId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| UserId | Текстовый | Идентификатор пользователя, внешний ключ, обязательное поле |
| DataCategoryId | Числовой | Идентификатор категории данных, внешний ключ, обязательное поле |
| ActionCategoryId | Числовой | Идентификатор категории действий, внешний ключ, обязательное поле |
| ItemId | Числовой | Идентификатор данных, обязательное поле |

Составлено автором

В таблице «ActionCategories» служебная таблица, хранящая категории действий. Поля, их типы и назначение представлены в таблице 2.10.

**Таблица 2.10 – Структура полей таблицы «ActionCategories»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| ActionCategoryId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Action | Текстовый | Категория действия, обязательное поле |

Составлено автором

Таблица «DataReferences» содержит зависимости данных. Поля, их типы и назначение представлены в таблице 2.11.

**Таблица 2.11 – Структура полей таблицы «ActionCategories»**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| 1 | 2 | 3 |
| DataReferenceId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |

**Окончание таблицы 2.11**

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| CityId | Числовой | Идентификатор города, внешний ключ, обязательное поле |
| AttractionId | Числовой | Идентификатор роли достопримечательности, внешний ключ, обязательное поле |

Составлено автором

Таким образом, произведено проектирование базы данных. Логическая схема базы данных представлена в приложении Б.