**2 Проектирование программного продукта**

**2.1 Постановка задачи**

В процессе выполнения дипломного проекта необходимо разработать приложение «Твой Витебск», представляющее актуальную информацию о жизни города Витебска. 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.

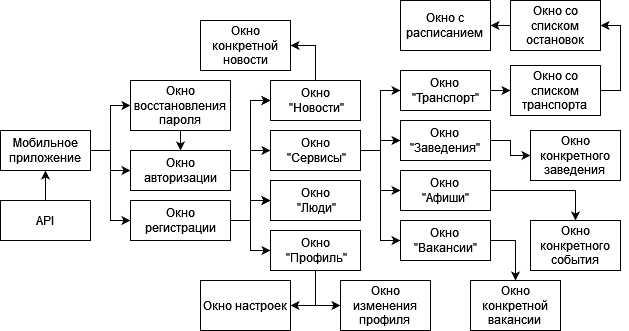
Структура серверного модуля приложения представлена на рисунке 2.1.

****

**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. Среда позволяет разрабатывать как [консольные](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологий «[Windows.Forms](https://ru.wikipedia.org/wiki/Windows_Forms" \o "Windows Forms)» и WPF, а также [веб-сайты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](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 [веб-службы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) [3],
* «Visual Studio Code» – это редактор исходного кода. Он имеет многоязычный интерфейс пользователя и поддерживает ряд языков программирования, подсветку синтаксиса, рефакторинг, отладку, навигацию по коду, поддержку «Git» и другие возможности [4],
* «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 [5],
* SQL – [декларативный](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), применяемый для создания, модификации и управления данными в [реляционной базе данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), управляемой соответствующей [системой управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [6],
* «Xamarin.Forms» –платформа пользовательского интерфейса с открытым кодом. С помощью «Xamarin.Forms» разработчики могут создавать приложения для «Android», iOS и UWP на основе общей базы кода. «Xamarin.Forms» позволяет разработчикам создавать разметку пользовательских интерфейсов с использованием XAML, а также бизнес-логику с помощью C# [7],
* «ASP.NET Core 5.0 MVC» – многофункциональная платформа для создания веб-приложений и API-интерфейсов с помощью структуры проектирования «Model-View-Controller» [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 [10].
* «ADO.NET Entity Framework» – объектно-ориентированная технология доступа к данным [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», «News», «CafeTypes», «Cafes», «PosterTypes», «Posters», «Vacancies», «Services».

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 |
| PasswordHash | Бинарный | Хэш пароля, обязательное поле |
| PasswordSalt | Бинарный | Хэш добавочной строки к паролю, используемая для усложнения хэша пароля, обязательное поле |
| FirstName | Текстовый | Имя, обязательное поле |
| LastName | Тектовый | Фамилия, обязательное поле |
| PhoneNumber | Текстовый | Номер телефона |
| RoleId | Числовой | Идентификатор роли пользователя, внешний ключ, обязательное поле |
| IsVisible | Логический | Видимость пользователя для других пользователей, обязательное поле |

Составлено автором

Таблица «News» содержит новостные статьи. Поля, их типы и назначение представлены в таблице 2.3.

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

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| NewsId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Title | Текстовый | Заголовок новости, обязательное поле |
| Description | Текстовый | Тело новости, обязательное поле |
| ExternalLink | Числовой | Ссылка на источник |

Составлено автором

Таблица «CafeTypes» содержит виды заведений питания. Поля, их типы и назначение представлены в таблице 2.4.

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

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| CafeTypeId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Name | Текстовый | Вид заведения, обязательное поле |

Составлено автором

Таблица «Cafes» отражает список заведений питания. Поля, их типы и назначение представлены в таблице 2.5.

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

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| 1 | 2 | 3 |
| CafeId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| CafeTypeId | Текстовый | Идентификатор вида заведения, внешний ключ, обязательное поле |
| Title | Текстовый | Название заведения, обязательное поле |
| Description | Текстовый | Описание заведения, обязательное поле |
| WorkingTime | Текстовый | Рабочие часы, обязательное поле |

**Окончание таблицы 2.5**

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Address | Текстовый | Адрес заведения, обязательное поле |
| ExternalLink | Числовой | Ссылка на источник |

Составлено автором

В таблице «PosterTypes» содержатся виды афиш. Поля, их типы и назначение представлены в таблице 2.6.

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

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| PosterTypeId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Name | Текстовый | Вид афиши, обязательное поле |

Составлено автором

Таблица «Posters» отражает афиши предстоящих событий. Поля, их типы и назначение представлены в таблице 2.7.

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

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| PosterId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| PosterTypeId | Текстовый | Идентификатор вида афиши, внешний ключ, обязательное поле |
| Title | Текстовый | Заголовок афиши, обязательное поле |
| Description | Текстовый | Описание афиши, обязательное поле |
| DateTime | Текстовый | Время проведения события, обязательное поле |
| Address | Текстовый | Адрес события, обязательное поле |
| ExternalLink | Числовой | Ссылка на источник |

Составлено автором

Таблица «Vacancies» содержит список открытых вакансий. Поля, их типы и назначение представлены в таблице 2.8.

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

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| VacancyId | Числовой | Первичный ключ, уникальный идентификатор, присваивается системой автоматически, обязательное поле |
| Title | Текстовый | Заголовок вакансии, обязательное поле |
| Description | Текстовый | Описание вакансии, обязательное поле |
| Requirements | Текстовый | Требования к соискателю, обязательное поле |
| Conditions | Текстовый | Условия работы, обязательное поле |
| Salary | Текстовый | Предполагаемая заработная плата, обязательное поле |
| CompanyName | Текстовый | Наименование организации работодателя, обязательное поле |
| Contacts | Текстовый | Контактные данные, обязательное поле |
| Address | Текстовый | Адрес организации, обязательное поле |
| PublishDate | Дата | Дата публикации, присваивается системой автоматически, обязательное поле |

Составлено автором

В служебной таблице «Services» содержатся названия сервисов приложения. Поля, их типы и назначение представлены в таблице 2.9.

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

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

Составлено автором

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