

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий
Кафедра Информационные системы и технологии
Специальность 1–40 01 01 «Программное обеспечение информационных технологий»
Специализация Программирование интернет-приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«Разработка базы данных для винного магазина»

Выполнил студент Пицуха Ярослав Анатольевич
(Ф.И.О.)
Руководитель проекта пр.-ст. Бондарчик Е.Н
(учен. степень, звание, должность, Ф.И.О., подпись)
Заведующий кафедрой к.т.н., доц. Смелов В.В.
(учен. степень, звание, должность, Ф.И.О., подпись)
Консультант: пр.-ст. Бондарчик Е.Н
(учен. степень, звание, должность, Ф.И.О., подпись)
Нормоконтролер: пр.-ст. Бондарчик Е.Н
(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой _____

Минск 2019

Реферат

Пояснительная записка курсового проекта содержит 32 страницы пояснительной записки, 13 таблиц, 32 иллюстрации, 9 источников литературы, 3 приложения.

C#, WPF, MICROSOFT SQL SERVER, БАЗА ДАННЫХ, SQL, T-SQL, ADO.NET

Целью курсового проекта является создание базы данных для винного магазина.

В первой главе проводится аналитический обзор прототипов по тематике курсового проекта и содержится описание технологий, использованных во время выполнения проекта.

Вторая глава посвящена процессу проектирования модели базы данных.

В третьей главе описывается процесс разработки, принципы функционирования и назначение созданных компонент проекта.

В четвёртой главе описаны процедуры для работы с XML.

В пятой главе содержатся сведения об использованной технологии.

Шестая глава посвящена тестированию производительности приложения и самого приложения.

В седьмой главе описывается руководство пользователя

В заключении приведены итоги курсового проектирования.

				КП 00.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пищу́ха Я. А.			Реферат			Лит.	Лист	Листов	
Провер.	Бондарчи́к Е.Н.								2	1
							БГТУ 71191005, 2019			
Н. контр.	.									
Утверд.										

Содержание

Введение.....	6
1 Постановка задачи	7
1.1 Обзор прототипов	7
1.2 Анализ прототипов	9
1.3 Описание используемых технологий.....	9
2 Проектирование модели базы данных.....	11
3 Разработка необходимых объектов.....	12
3.1 Таблицы	12
3.2 Пользователи	16
3.3 Процедуры	16
4 Описание процедур импорта и экспорта данных.....	19
4.1 Процедура импорта данных из XML-файла	19
4.2 Процедура экспорта данных в формат XML	19
5 Описание технологии	20
6 Тестирование.....	21
6.1 Тестирование производительности базы данных.....	21
6.2 Тестирование работоспособности приложения	22
7 Руководство пользователя	25
Заключение.....	31
Список литературных источников.....	32
Приложение А.....	33
Приложение Б	34
Приложение В.....	40

				КП 00.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пицуха Я.А.			Содержание			Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.								3	1
							БГТУ 71191005, 2019			
Н. контр.	.									
Утверд.										

Введение

Данные всегда играли важную роль в жизни людей и в разные времена их хранили по-разному. В наше время всё больше информации хранится в цифровом виде и для хранения такой информации отлично подходят различные базы данных. Использование базы данных упрощает управление и хранение данных. Одним из примеров успешного внедрения баз данных является использование их различными магазинами.

Сейчас есть множество удобных приложений, позволяющих без особых усилий совершать покупки, с помощью таких приложений обычно можно совершить полный цикл покупки, начиная от выбора товара по необходимым параметрам и заканчивая оформлением заказа.

Целью моего курсового проекта является создание базы данных винного магазина.

Основными задачами курсовой работы являются:

- провести аналитический обзор литературы;
- спроектировать базу данных;
- реализовать функциональность базы данных;
- провести тестирование используемой технологии в базе данных;
- разработать приложение для работы с базой данных;
- написать руководство пользователя.

В соответствии с заданием курсового проекта для проектирования базы данных винного магазина используется система управления базами данных Microsoft SQL Server.

Также в задачу входит создание удобного пользовательского интерфейса для взаимодействия с базой данных магазина. Для этого создаётся приложение, которое будет облегчать пользователю работу с необходимыми данными.

Для разработки приложения использовалась технология WPF и язык программирования C#. Для взаимодействия приложения с базой данных использовалась технология ADO.NET.

				КП 00.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пицуха Я.А.			Введение			Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.							6	1	
							БГТУ 71191005, 2019			
Н. контр.	.									
Утверд.										

1 Постановка задачи

В соответствии с заданием курсового проекта следует не только создать базу данных, но и разработать программное средство, которое должно в полной мере показать возможности базы данных. Для того чтобы сформировать окончательные требования к проектируемому программному средству сначала рассмотрим прототипы из той же области.

1.1 Обзор прототипов

На сегодняшний день существует множество сайтов, которые дают своим клиентам возможность совершать покупки различной винной продукции. Рассмотрим примеры web-приложений подобных сайтов.

WineStyle[1]

Одним из сайтов для покупки вина является WineStyle. На этом сайте можно не только выбрать, но и приобрести понравившееся вино, а также некоторые иные виды алкоголя. Подробная информация о товарах, широкий ассортимент продукции и наличие акций и скидок, делают этот сайт популярным у пользователей.

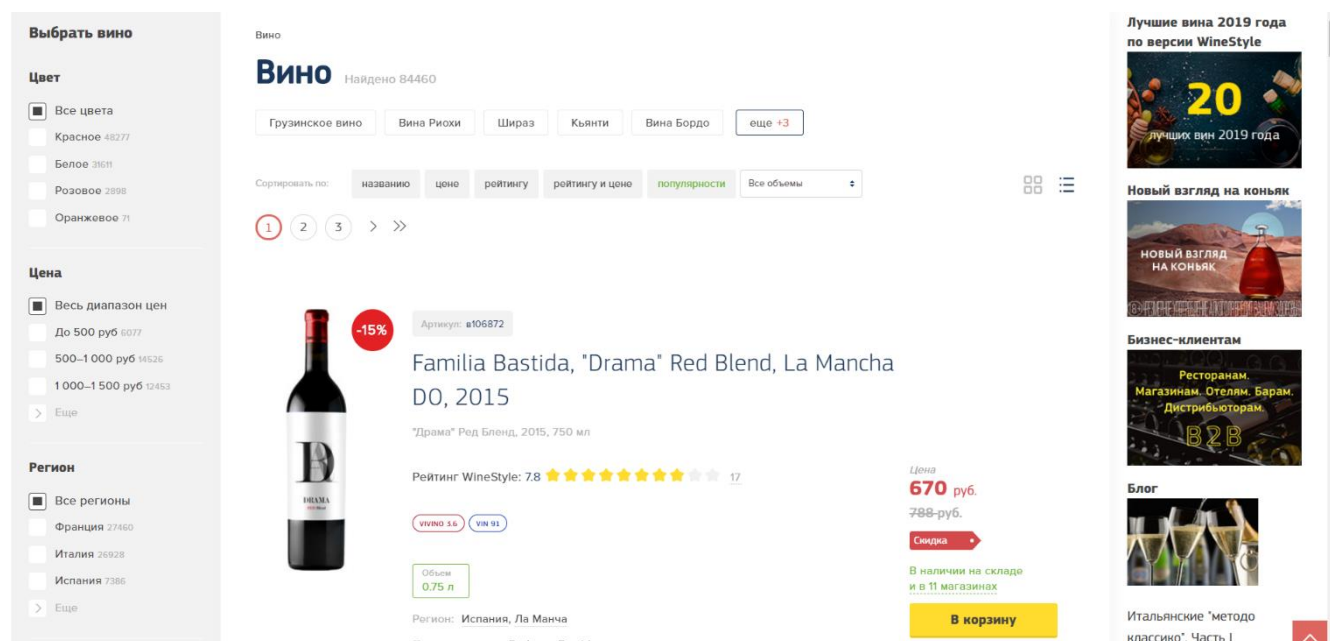


Рисунок 1.1 – Интерфейс сайта WineStyle

Goodwin[2]

Данный сайт имеет простой, но при этом очень удобный и понятный интерфейс. Сайт имеет множество видов вин, а также имеются другие алкогольные напитки. Особенностью сайта является наличие большого выбора сыров. При выборе продукции пользователь может фильтровать каталог товаров по очень

				КП 01.00.ПЗ								
	ФИО	Подпись	Дата									
Разраб.	Пицуха Я.А.			1 Постановка задачи				Лит.	Лист	Листов		
Провер.	Бондарчик Е.Н.										7	4
Н. контр.	.											
Утверд.												
											БГТУ 71191005, 2019	

многим параметрам. Так же сайт предлагает пользователям множество акций и специальных предложений при покупке. Из всех рассмотренных мною сайтов для продажи вина, этот мне показался самым удобным.

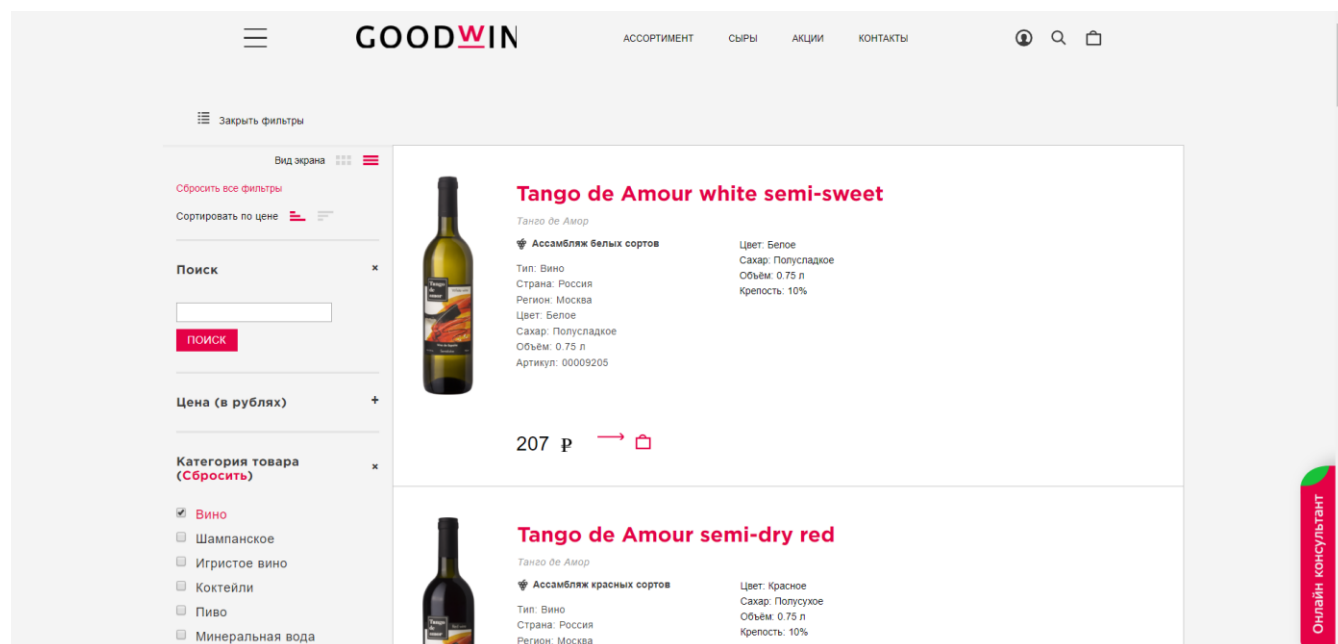


Рисунок 1.2 – Интерфейс сайта Goodwin

Ароматный мир[3]

Ароматный мир – это крупнейшая в России сеть специализированных винмаркетов. В связи с этим при заходе на сайт, первое что бросается в глаза, это огромный выбор различной продукции. Каталог кроме различных видов алкогольной продукции содержит также некоторые виды продуктов питания. Имеются удобные фильтры и сортировки на страницах выбора продукции. Дизайн сайта, несмотря на множество элементов, понятен и привлекателен. Большим плюсом сайта является наличие обширного личного кабинета пользователя.

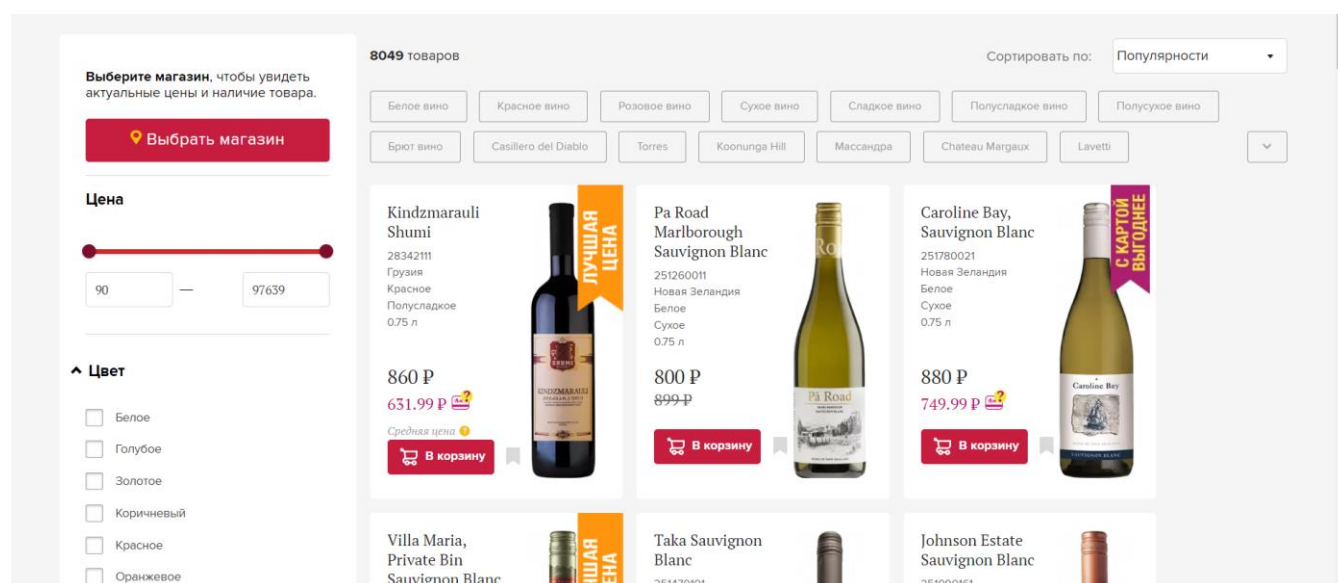


Рисунок 1.3 – Интерфейс сайта Ароматный мир

1.2 Анализ прототипов

Проанализировав прототипы сайтов, можно отметить ожидаемую схожесть в их функциональности и интерфейсе.

Исходя из полученных результатов, можно сформулировать основные функциональные требования курсовой работы:

- регистрация и авторизация клиентов;
- авторизация администратора;
- поиск товаров по параметрам;
- наличие различных сортировок товара;
- просмотр сведений о товаре;
- занесение пользователем выбранных товаров в корзину;
- оформление заказа пользователем;
- указание и изменение различной информации пользователем в своём личном кабинете;
- добавление администратором нового вина;
- редактирование информации о продуктах;

Данные требования будут реализованы в базе данных в виде хранимых процедур, которые описаны в главе 3 данной пояснительной записки.

1.3 Описание используемых технологий

СУБД существует огромное множество: Oracle, MS SQL Server, NoSQL, MySql и так далее. Для организации работы с базой данных в данной работе было решено использовать одну из наиболее популярных СУБД «Microsoft SQL Server» [4], так как она предоставляет необходимые возможности оперирования объектами базы данных.

В качестве интерфейса прикладного программирования был выбран API-интерфейс — Windows Presentation Foundation (WPF) [5], платформа уровня представления для построения графических интерфейсов. Основывается на векторной системе визуализации и ориентирована на разработку клиентских Windows приложений, базирующихся на технологии Microsoft.NET.

WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление. Что является преимуществом перед более ранней технологией создания пользовательских интерфейсов — Windows Forms.

Для работы с WPF использовался объектно-ориентированный язык программирования с C-подобным синтаксисом — C#, разработанный для создания приложений на платформе Microsoft .NET Framework [6].

Чтобы осуществлять связь между базой данных и приложением на C# необходим посредник. И именно таким посредником является технология ADO.NET [7]. ADO.NET – интерфейс прикладного уровня Microsoft (набор классов, предоставляющих службы доступа к данным). Является не развитием более ранней

технологии ADO, а самостоятельной технологией, частью фреймворка .NET. Для обеспечения защиты паролей клиентов применяется технология хеширования.

Также в процессе выбора технологии была рассмотрена технология Full-text search, которая позволяет сильно расширить возможности.

По сведениям из документации [8] в полнотекстовый индекс включается один или несколько символьных столбцов таблицы. Эти столбцы могут содержать любой из следующих типов данных: char, varchar, nchar, nvarchar, text, ntext, image, xml или varbinary(max) и FILESTREAM. Каждый полнотекстовый индекс индексирует один или несколько столбцов таблицы, а каждому столбцу может соответствовать определенный язык.

Полнотекстовые запросы выполняют лингвистический поиск в текстовых данных в полнотекстовых индексах путем обработки слов и фраз в соответствии с правилами конкретного языка, например английского или японского. Полнотекстовые запросы могут включать простые слова и фразы или несколько форм слова или фразы. Полнотекстовый запрос возвращает все документы, которые содержат как минимум одно совпадение (известное также как попадание). Совпадение возникает в том случае, когда целевой документ содержит все термины, указанные в полнотекстовом запросе, и соответствует всем остальным условиям поиска, например расстояние между совпадающими терминами.

2 Проектирование модели базы данных

Для реализации поставленной задачи была создана база данных WineStore. Диаграмма структуры полученной базы данных, разработанной в СУБД «Microsoft SQL Server», представлена в приложении А.

Для базы данных было разработано 11 таблиц, которые связаны друг с другом внешними ключами.

Таблица USER хранит данные о клиентах.

Таблица SESSION содержит информацию о том, находится ли пользователь в системе в данный момент времени.

Таблица ABOUT_USER хранит подробную информацию о пользователях, необходимую для упрощения процесса оформления заказа.

Таблица CARD хранит информацию о данных платёжной карты пользователя.

Таблица WINE содержит информацию о всех винах каталога магазина.

Таблица BASKET необходима для связи таблицы USER и WINE.

Таблица ORDER_HISTORY содержит информацию всех совершённых заказов магазина.

Таблица ORDER_INFORMATION хранит подробную информацию заказов.

Таблица ORDER связывает таблицы ORDER_HISTORY и WINE.

Таблица SHOP хранит информацию о магазине.

Таблица SHOP_FAQS хранит вопросы и ответы на них, необходимые для соответствующего раздела магазина.

Создание перечисленных таблиц с установлением внешних связей описано в главе 3.

				КП 02.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пицуха Я.А.			2 Проектирование модели базы данных			Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.								11	1
							БГТУ 71191005, 2019			
Н. контр.	.									
Утверд.										

3 Разработка необходимых объектов

3.1 Таблицы

Таблицы являются неотъемлемой частью любой реляционной базы данных. Краткая характеристика каждой из таблиц была предоставлена в разделе 2, а код их создания можно увидеть в Приложении Б. Ниже мы рассмотрим каждую таблицу подробнее.

Таблица USER состоит из пяти столбцов:

- ID – хранит идентификатор пользователя;
- LOGIN – логин пользователя;
- PASSWORD – пароль пользователя;
- ABOUT – идентификатор подробной информации о пользователе, внешний ключ к таблице ABOUT_USER;
- IS_ACTIVE – идентификатор на информацию о сессии пользователя, внешний ключ к таблице SESSION.

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	<input type="checkbox"/>
	LOGIN	nvarchar(30)	<input type="checkbox"/>
	PASSWORD	nvarchar(30)	<input type="checkbox"/>
	ABOUT	int	<input checked="" type="checkbox"/>
	IS_ACTIVE	int	<input checked="" type="checkbox"/>

Рисунок 3.1 – Описание таблицы USER

Таблица SESSION состоит из трёх столбцов:

- ID – хранит идентификатор сессии;
- LOGIN – логин пользователя, которому принадлежит данная сессия;
- ACTIVE – информацию о том, активен ли пользователь в данный момент времени.

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	<input type="checkbox"/>
	LOGIN	nvarchar(30)	<input type="checkbox"/>
	ACTIVE	bit	<input checked="" type="checkbox"/>

Рисунок 3.2 – Описание таблицы SESSION

				КП 03.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пицуха Я.А.			3 Разработка необходимых объектов			Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.								12	7
							БГТУ 71191005, 2019			
Н. контр.	.									
Утверд.										

Таблица ABOUT_USER состоит из семи столбцов:

- ID – хранит идентификатор информации пользователя;
- FIRST_NAME – имя пользователя;
- LAST_NAME – фамилия пользователя;
- STREET_ADDRESS – адрес доставки пользователя;
- CITY – город доставки пользователя;
- PHONE_NUMBER – номер телефона пользователя;
- CARD_INFO – информация о платёжной карте пользователя, внешний ключ к таблице CARD.

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	<input type="checkbox"/>
	FIRST_NAME	nvarchar(50)	<input type="checkbox"/>
	LAST_NAME	nvarchar(50)	<input type="checkbox"/>
	STREET_ADDRESS	nvarchar(50)	<input type="checkbox"/>
	CITY	nvarchar(50)	<input type="checkbox"/>
	PHONE_NUMBER	nvarchar(50)	<input type="checkbox"/>
	CARD_INFO	int	<input checked="" type="checkbox"/>

Рисунок 3.3 – Описание таблицы ABOUT_USER

Таблица CARD состоит из четырёх столбцов:

- ID – хранит идентификатор платёжной карты;
- CARD_NUMBER – номер карты;
- DATE – дата, до которой карта действительна;
- CVC – CVC код карты.

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	<input type="checkbox"/>
	CARD_NUMBER	int	<input checked="" type="checkbox"/>
	DATE	nvarchar(5)	<input checked="" type="checkbox"/>
	CVC	int	<input checked="" type="checkbox"/>

Рисунок 3.4 – Описание таблицы CARD

Таблица WINE состоит из одиннадцати столбцов:

- ID – хранит идентификатор вина;
- PRODUCT_CODE – код продукта;
- TITLE – название вина;
- VOLUME – объём бутылки;
- PRICE – стоимость продукта;
- VINTAGE – возраст бутылки с вином;
- AVAILABLE – количество бутылок данного вина в магазине;
- TYPE – тип вина;
- COUNTRY – страна производства вина;
- COTEGORY – информация к какой категории относится данное вино;
- FOTO – ссылка на фото продукта.

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	<input type="checkbox"/>
	PRODUCT_CODE	nvarchar(15)	<input type="checkbox"/>
	TITLE	nvarchar(200)	<input type="checkbox"/>
	VOLUME	int	<input type="checkbox"/>
	PRICE	decimal(10, 2)	<input type="checkbox"/>
	VINTAGE	int	<input type="checkbox"/>
	AVAILABLE	int	<input type="checkbox"/>
	TYPE	nvarchar(20)	<input type="checkbox"/>
	COUNTRY	nvarchar(30)	<input type="checkbox"/>
	COTEGORY	nvarchar(30)	<input type="checkbox"/>
	FOTO	nvarchar(100)	<input type="checkbox"/>

Рисунок 3.5 – Описание таблицы WINE

Таблица BASKET состоит из трёх столбцов:

- ID – хранит идентификатор корзины;
- USER – идентификатор пользователя, которому принадлежит данная корзина, внешний ключ к таблице USER;
- WINE – идентификатор вина, которое находится в данной корзине, внешний ключ к таблице WINE.

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	<input type="checkbox"/>
	[USER]	int	<input checked="" type="checkbox"/>
	WINE	int	<input checked="" type="checkbox"/>

Рисунок 3.6 – Описание таблицы BASKET

Таблица ORDER_HISTORY состоит из шести столбцов:

- NUMBER – хранит идентификатор заказа;
- USER – идентификатор пользователя который сделал заказ, внешний ключ к таблице USER;
- DATE – дата и время заказа;
- PRODUCTCOUNT – количество бутылок в заказе;
- PRICE – общая цена заказа;
- INFO – идентификатор подробной информации о заказе, внешний ключ к таблице ORDER_INFORMATION.

	Имя столбца	Тип данных	Разрешить знач...
🔑	NUMBER	int	<input type="checkbox"/>
	[USER]	int	<input checked="" type="checkbox"/>
	DATE	datetime	<input type="checkbox"/>
	ProductCount	int	<input checked="" type="checkbox"/>
	Price	int	<input type="checkbox"/>
	Info	int	<input checked="" type="checkbox"/>

Рисунок 3.7 – Описание таблицы ORDER_HISTORY

Таблица ORDER_INFORMATION состоит из девяти столбцов:

- ID – хранит идентификатор информации о заказе;
- FIRST_NAME – имя на которое сделан заказ;
- LAST_NAME – фамилия на которую сделан заказ;
- STREET_ADDRESS – адрес доставки;
- CITY – город;
- PHONE_NUMBER – номер телефона, указанного при оформлении;
- CARD_NUMBER – номер карты, с помощью которой оформлен заказ;
- DATE – дата, до которой карта действительна;
- CVC – CVC код карты.

	Имя столбца	Тип данных	Разрешить знач...
PK	ID	int	<input type="checkbox"/>
	FIRST_NAME	nvarchar(50)	<input type="checkbox"/>
	LAST_NAME	nvarchar(50)	<input type="checkbox"/>
	STREET_ADDRESS	nvarchar(50)	<input type="checkbox"/>
	CITY	nvarchar(50)	<input type="checkbox"/>
	PHONE_NUMBER	nvarchar(50)	<input type="checkbox"/>
	CARD_NUMBER	int	<input type="checkbox"/>
	DATE	nvarchar(5)	<input type="checkbox"/>
	CVC	int	<input type="checkbox"/>

Рисунок 3.8 – Описание таблицы ORDER_INFORMATION

Таблица ORDER состоит из трёх столбцов:

- NUMBER – хранит идентификатор связи в заказе;
- ORDER_NUMBER – идентификатор заказа, внешний ключ к таблице ORDER_HISTORY;
- WINE – идентификатор вина из заказа, внешний ключ к таблице WINE.

	Имя столбца	Тип данных	Разрешить знач...
PK	NUMBER	int	<input type="checkbox"/>
	ORDER_NUMBER	int	<input checked="" type="checkbox"/>
	WINE	int	<input checked="" type="checkbox"/>

Рисунок 3.9 – Описание таблицы ORDER

Таблица SHOP состоит из шести столбцов:

- ID – хранит идентификатор магазина;
- ADDRESS – адрес магазина;
- TELEPHON_NUMBER – номер телефона магазина;
- EMAIL – электронная почта;
- DESCRIPTION – описание;
- ALL_WINE – количество всего вина в магазине.

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	<input type="checkbox"/>
	ADDRESS	nvarchar(100)	<input type="checkbox"/>
	TELEPHON_NUMBER	nvarchar(20)	<input type="checkbox"/>
	Email	nvarchar(20)	<input type="checkbox"/>
	DESCRIPTION	nvarchar(1000)	<input checked="" type="checkbox"/>
	ALL_WINE	int	<input checked="" type="checkbox"/>

Рисунок 3.10 – Описание таблицы SHOP

Таблица SHOP_FAQS состоит из четырёх столбцов:

- ID – хранит идентификатор вопроса;
- QUESTION – вопрос;
- ANSWER – ответ;
- SHOP_NAME – идентификатор магазина, внешний ключ к таблице SHOP.

	Имя столбца	Тип данных	Разрешить знач...
🔑	ID	int	<input type="checkbox"/>
	question	nvarchar(400)	<input type="checkbox"/>
	answer	nvarchar(400)	<input type="checkbox"/>
	SHOP_NAME	int	<input checked="" type="checkbox"/>

Рисунок 3.11 – Описание таблицы SHOP_FAQS

3.2 Пользователи

Пользователь базы данных – это физическое или юридическое лицо, которое имеет доступ к базе данных и пользуется услугами информационной системы для получения информации. Пользователи базы данных получают права доступа для чтения, вставки, обновления и удаления конкретных объектов.

Администратор винного магазина наделён привилегией на выполнение хранимых процедур добавления и изменения новых товаров и хранимых процедур для работы с XML.

Клиенту разрешено выполнять только процедуры для авторизации и регистрации, процедуры для получения предоставленной ему информации из базы данных, процедуры для добавления товаров и заказов, а также процедуру для его работы со своим личным кабинетом.

Скрипты для создания логинов и пользователей базы данных, а также выделение привилегий на осуществление определённых операций с базой данных представлены в приложении Б.

3.3 Процедуры

Использование хранимых процедур позволяет ограничить либо вообще исключить непосредственный доступ пользователей к таблицам базы данных, оставив пользователям только разрешения на выполнение хранимых процедур, обеспечивающих косвенный и строго регламентированный доступ к данным.

Всего было разработано 28 процедур для работы с данными. Краткое описание процедур описано далее.

Пример создания хранимой процедуры `sp_AlterWines`, которая изменяет заданные параметры товара представлен в приложении Б.

Данная процедура принимает входной параметр `Code` (код продукта), `Price` (новое значение цены товара) и `Available` (новое значение количества данного продукта). Тело процедуры находит по заданному коду нужное вино и обновляет у него параметры цены и количества.

`sp_UserVerification` – процедура для проверки на существование заданного пользователя;

`sp_UserLogout` – процедура для закрытия сессии пользователя;

`sp_InsertUser` – процедура для добавления нового пользователя;

`sp_SelectALLWines` – процедура для выборки всех вин;

`sp_SelectAllWinesSortedPrice` – процедура для выборки всех вин, отсортированных по цене;

`sp_SelectAllWinesSortedName` – процедура для выборки всех вин, отсортированных по названию;

`sp_SelectAllWinesSortedVintage` – процедура для выборки всех вин, отсортированных по возрасту;

`sp_SelectWines` – процедура для выборки всех вин по заданным параметрам;

`sp_AddWines` – процедура добавления нового вина в каталог;

`sp_AlterWines` – процедура изменения параметров выбранного вина;

`sp_AddWinesToBasket` – процедура добавления вина в корзину;

`sp_SelectALLWinesToBasket` – процедура выборки всех вин добавленных в корзину;

`sp_RemoveWineFromBasket` – процедура удаления выбранного вина из корзины;

`sp_CreateOrder` – процедура создания нового заказа;

`sp_InsertOrderInformation` – процедура добавления подробной информации о сделанном заказе;

`sp_SelectALLOrders` – процедура выборки всех заказов пользователя;

`sp_ImportWineFromXML` – процедура создания нового вина по параметрам из XML файла;

`sp_ExportWineToXML` – процедура экспорта всех вин в формате XML;

`sp_SelectShippingInformation` – процедура выборки информации о пользователе;

`sp_InsertToAboutUser` – процедура добавления новой подробной информации о пользователе;

`sp_UpdateShippingInformation` – процедура обновления информации о пользователе;

`sp_SelectPaymentInformation` – процедура выборки информации о платёжной карте пользователя;

`sp_InsertToCard` – процедура добавления информации о новой платёжной карте;

`sp_UpdatePaymentInformation` – процедура обновления платёжной информации;

`sp_SelectAllFaqs` – процедура выборки всех вопросов и ответов магазина;

sp_SelectFromFaqs – процедура для поиска и выборки вопросов и ответов на основе упоминания в входном параметре;

sp_SelectShopInfo – процедура для выборки информации о магазине;

sp_filling_out – процедура для занесения ста тысяч новых пользователей.

Все скрипты хранимых процедур приложены в отдельном файле в корне директории прилагаемого диска.

4 Описание процедур импорта и экспорта данных

База данных обычно имеет не самостоятельную ценность, является частью информационной системы. Независимо от того, как устроена эта система, на противоположном от БД конце находится интерфейс взаимодействия с пользователем, и задача программиста предоставить простой и понятный способ работы с хранящимися в БД данными и объектами.

При всей своей отлаженности и очевидности, классический способ хранения и представления объектов развитой структуры имеет и вполне определенные недостатки и может вызывать проблемы, с которыми сталкивался любой разработчик, пытавшийся реализовать таким способом достаточно сложную систему. В некоторых ситуациях, решить эти проблемы позволяет хранение объекта в виде XML.

4.1 Процедура импорта данных из XML-файла

Для импорта используется стандартная функция, входящая в набор OLE DB – OPENROWSET, в которую передаются параметры о типе импортируемых данных и пути файла, где они находятся. С помощью процедуры sp_ImportWineFromXML администратор может добавить новое вино в каталог магазина из XML-файла. Листинг данной процедуры можно увидеть в приложении Б.

4.2 Процедура экспорта данных в формате XML

Процедура sp_ExportWineToXML по выходному параметру возвращает данные о всех винах каталога магазина в формате XML, которые потом отображаются в приложении. Её реализацию можно увидеть в приложении Б.

				КП 04.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пицуха Я. А.			4 Описание процедур импорта и экспорта данных			Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.							1	1	
							БГТУ 71191005, 2019			
Н. контр.	.									
Утверд.										

5 Описание технологии

Полнотекстовый поиск в SQL Server позволяет пользователям и приложениям выполнять полнотекстовые запросы к символьным данным в таблицах SQL Server .

В полнотекстовый индекс включается один или несколько символьных столбцов таблицы. Каждый полнотекстовый индекс индексирует один или несколько столбцов таблицы, а каждому столбцу может соответствовать определенный язык.

Полнотекстовые запросы выполняют лингвистический поиск в текстовых данных в полнотекстовых индексах путем обработки слов и фраз в соответствии с правилами конкретного языка, например английского или японского. Полнотекстовые запросы могут включать простые слова и фразы или несколько форм слова или фразы. Полнотекстовый запрос возвращает все документы, которые содержат как минимум одно совпадение (известное также как попадание). Совпадение возникает в том случае, когда целевой документ содержит все термины, указанные в полнотекстовом запросе, и соответствует всем остальным условиям поиска, например расстояние между совпадающими терминами.

После добавления столбцов в полнотекстовый индекс, приложения и пользователи смогут выполнять полнотекстовые запросы к тексту из столбцов. Эти запросы могут вести поиск любых приведенных ниже элементов.

Одно или несколько конкретных слов или фраз (простое выражение);

Слова, начинающиеся заданным текстом, или фразы с такими словами (префиксные выражения);

Словоформы конкретного слова (производное выражение);

Слова или фразы, находящиеся рядом с другими словами или фразами (выражения с учетом расположения);

Синонимические формы конкретного слова (тезаурус);

Слова или фразы со взвешенными значениями (взвешенное выражение).

В полнотекстовых запросах не учитывается регистр букв. Например, поиск значения «Алюминий» или «алюминий» вернет одинаковые результаты.

Все полнотекстовые запросы используют небольшой набор предикатов Transact-SQL (CONTAINS и FREETEXT) и функций (CONTAINSTABLE и FREETEXTTABLE). Однако точная структура полнотекстовых запросов определяется целями поиска данного бизнес-сценария.

В процедуре sp_SelectWines поиск в таблице WINE осуществляется с помощью предиката CONTAINS, а также в процедуре sp_SelectFromFaqs поиск в таблице SHOP_FAQS построен на предикате FREETEXT. Листинг процедур приведён в приложении Б.

				КП 05.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пицуха Я. А.			5 Описание технологии			Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.								20	1
							БГТУ 71191005, 2019			
Н. контр.	.									
Утверд.										

6 Тестирование

6.1 Тестирование производительности базы данных

Для тестирования производительности была взята за основу таблица USER, которую мы заполнили 100001 строками. После этого был применён SELECT-запрос к данной таблице на столбец ID и при помощи стандартных средств IDE MS SQL Server Management Studio оценена цена выборки к таблице. Результат данной оценки запроса приведён на рисунке 6.1.

Clustered Index Scan (Clustered)	
Просмотр всего кластеризованного индекса или его части.	
Физическая операция	Clustered Index Scan
Логическая операция	Clustered Index Scan
Estimated Execution Mode	Row
Предполагаемая стоимость оператора	1,18291 (100%)
Предполагаемая стоимость операций ввода-вывода	1,07275
Estimated CPU Cost	0,110158
Предполагаемая стоимость поддерева	1,18291
Предполагаемое количество выполнений	1
Предполагаемое количество строк	100001
Предполагаемый размер строки	161 Б
Отсортировано	False
Идентификатор узла	0

Рисунок 6.1 – Оценка запроса к таблице USER с кластеризованным индексом

После проведения первоначальной оценки был построен некластеризованный индекс к таблице USER, создание которого можно увидеть в приложении Б, и проведена оценка такого же SELECT-запроса к таблице USER. Результаты, полученные во время оценки, представлены на рисунке 6.2.

				КП 06.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пищуха Я. А			6 Тестирование	Лит.		Лист		Листов	
Провер.	Бондарчик Е.Н.						21		4	
Н. контр.	.				БГТУ 71191005, 2019					
Утверд.										

Просмотр индекса (NonClustered)	
Просмотр всего некластеризованного индекса или его части.	
Физическая операция	Просмотр индекса
Логическая операция	Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Хранилище	RowStore
Фактическое количество строк	100003
Фактическое количество пакетов	0
Предполагаемая стоимость операций ввода-вывода	0,220903
Предполагаемая стоимость оператора	0,331063 (100%)
Предполагаемая стоимость поддеревя	0,331063
Estimated CPU Cost	0,11016
Предполагаемое количество выполнений	1
Количество выполнений	1
Предполагаемое количество строк	100003
Предполагаемый размер строки	11 Б
Фактическое число повторных привязок	0
Фактическое число сбросов на начало	0
Отсортировано	False
Идентификатор узла	0

Рисунок 6.2 – Оценка запроса к таблице с построенным некластеризованным индексом

По результатам проведённых оценок, можно сделать вывод, что после создания некластеризованного индекса:

- стоимость по параметру “Предполагаемая стоимость операций ввода-вывода” стала в 3 раза меньше;
- стоимость по параметру “Предполагаемая стоимость оператора” снизилась в 3 раза;
- стоимость по параметру “Предполагаемая стоимость поддеревя” уменьшилась в 3 раза.

Таким образом, постройка индекса к таблице была более чем оправдана, так как мы получили прирост производительности в 3 и более раза.

6.2 Тестирование работоспособности приложения

Тестирование приложения предназначено для выявления возможных ошибок при разработке и для того, чтобы исключить действия пользователя, которые могут повлечь за собой аварийное завершение программы.

Рассмотрим регистрацию и авторизацию пользователя. Если пользователь не заполнит какие-нибудь поля при регистрации или авторизации, то ему будет предложено заполнить все поля.

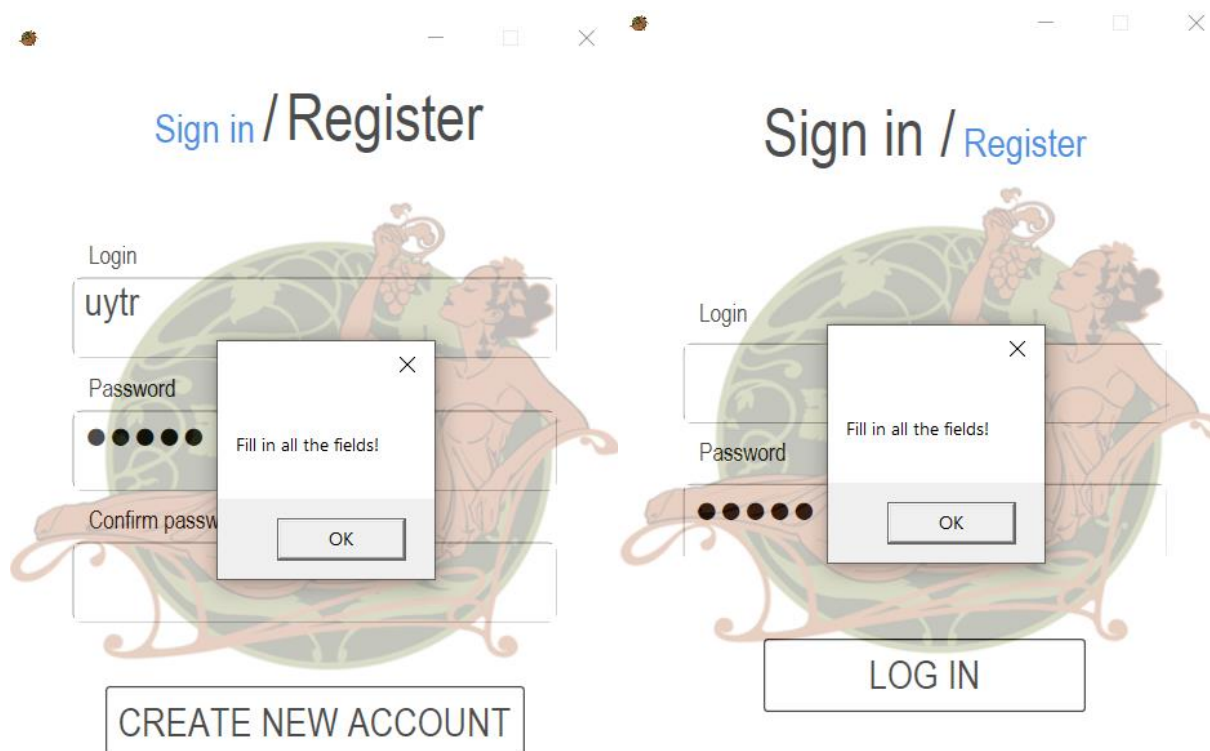


Рисунок 6.3 – Проверка заполнения полей при регистрации и авторизации

В окне авторизации пользователя, если пользователь введёт данные, которых нет в базе данных, то пользователю получит оповещение что такого пользователя нет.



Рисунок 6.4 – Проверка ошибки авторизации

При оформлении заказа также проводится контроль заполнения всех полей.

The screenshot shows the 'WineStore' checkout page. At the top, there are navigation buttons: 'WINES', 'BASKET', 'FAQs', and 'MY ACCOUNT'. Below these, there's a 'Shipping information' section with fields for 'First name', 'Last name', 'Street', 'City', 'Drog', and 'Phone'. A modal dialog box is open over the 'First name' field, displaying the message 'specify the First name parameter' with an 'OK' button. Below the shipping section is the 'Payment information' section with fields for 'Card number', 'Expiration date', and 'CVC'. The 'Card number' field contains '76543', 'Expiration date' contains '27/20', and 'CVC' contains '940'. At the bottom of the form is a 'COMPLETE' button.

Рисунок 6.5 – Проверка заполнения полей при оформлении нового заказа

Если пользователь ввёл не все данные при выборе вина по параметрам, то ему будет подсказано какое поле ещё не заполнено.

The screenshot shows the 'WineStore' browse page. On the left, there are filter sections for 'Sweet Wines', 'Fortified Wines', 'Type', 'Country', 'Vintage', 'Price', and 'Volume'. Each filter section has a 'show more' button. The main area is titled 'Browse Wines' and displays a grid of wine products. A modal dialog box is open over the grid, displaying the message 'specify the Country parameter' with an 'OK' button. The grid shows several wine bottles with their prices and 'BUY NOW' buttons. One bottle is labeled 'Gigondas Les Pallieres de Diable 600cl 2013' with a price of £312,00 and 'Available: 2'.

Рисунок 6.6 – Проверка на ввод всех параметров поиска

7 Руководство пользователя

Первое, что должен сделать пользователь после открытия приложения — это войти, если он не зарегистрирован, то он может перейти соответственно в окно регистрации и сначала зарегистрироваться.

Рисунок 7.1 – Окна регистрации или авторизации

После авторизации пользователь попадает на главное окно, на которой располагается информация о магазине.

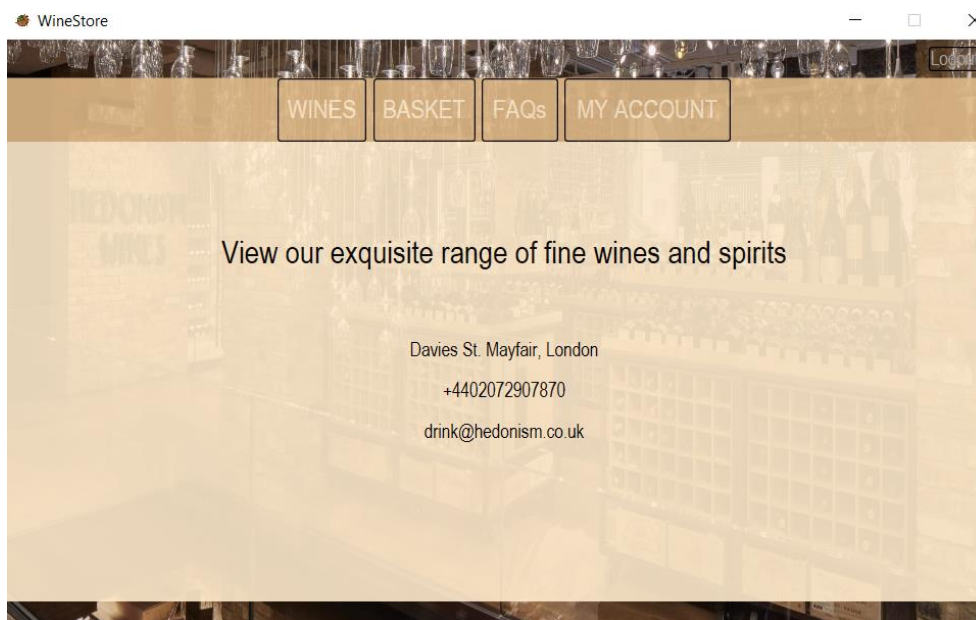


Рисунок 7.2 – Главное окно

Что бы перейти на вкладку с выбором товаров надо нажать на кнопку Wines.

				КП 07.00.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Пицуха Я. А.			7 Руководство пользователя	Лит.		Лист	Листов		
Провер.	Бондарчик Е.Н.						25	6		
					БГТУ 71191005, 2019					
Н. контр.	.									
Утверд.										

На экране отобразится наглядный список с информацией о всех винах, имеются кнопки для поиска по параметрам и сортировок. Что бы переместить вино в корзину, нужно нажать на кнопку **BUY NOW** возле выбранного вина.

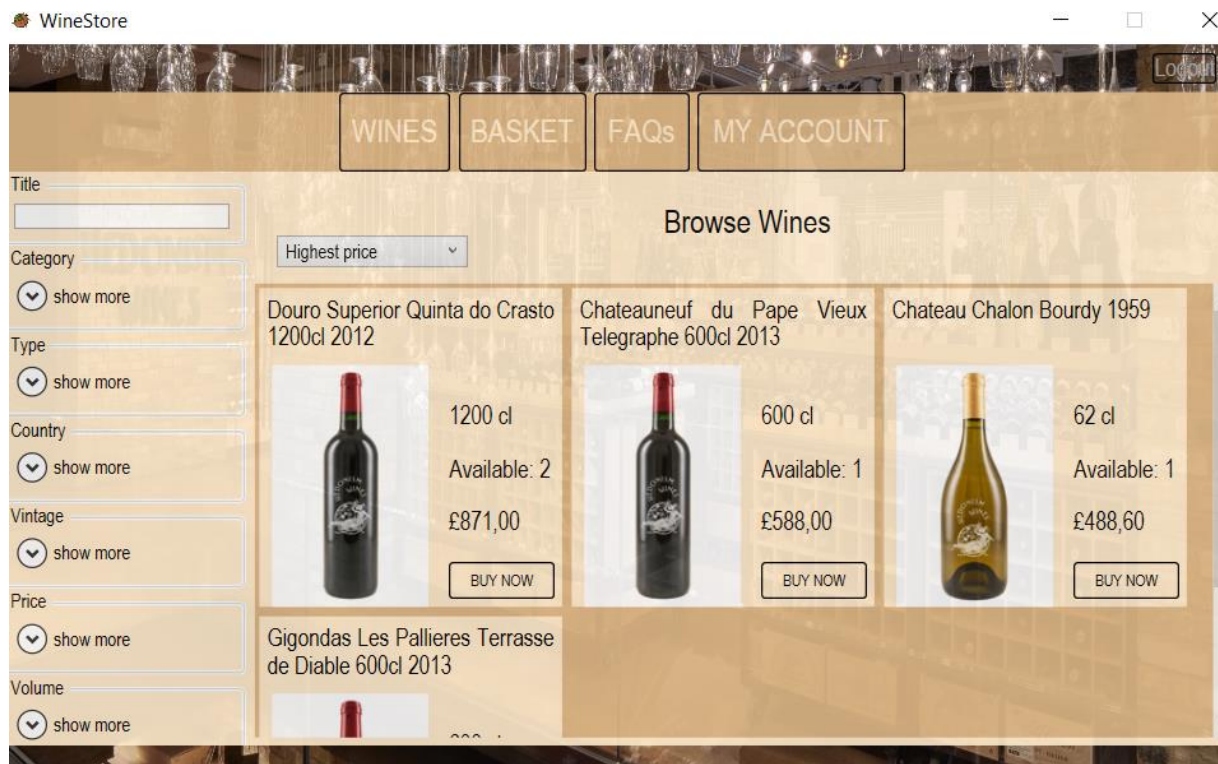


Рисунок 7.3 – Окно выбора вина

Нажав на вкладку **BASKET** откроется список вин в корзине, можно убрать вино из корзины, либо нажать на кнопку **CHECKOUT** и перейти к заполнению информации о заказе.

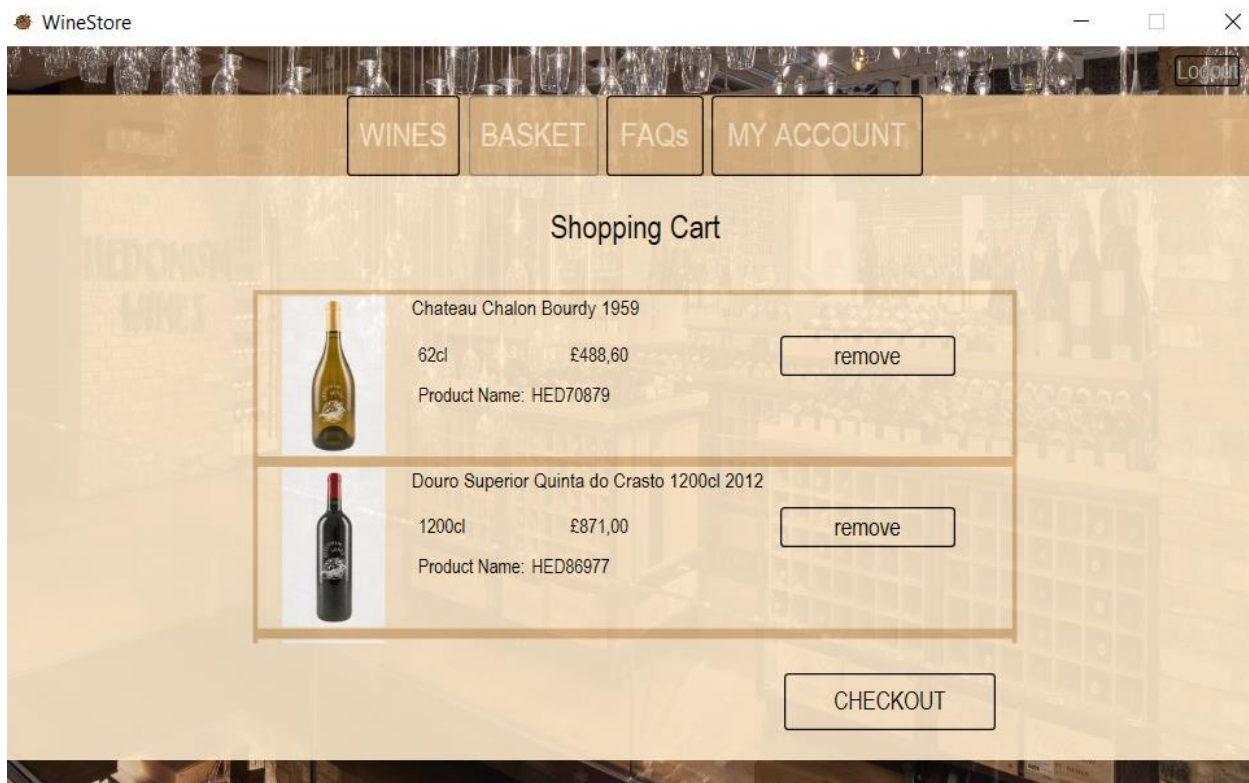


Рисунок 7.4 – Страница с корзиной товаров

После окончания заполнения информации следует нажать кнопку COMPLETE.

WineStore

WINES BASKET FAQs MY ACCOUNT

Shipping information

First name
Yaroslav

Last name
Pitsukha

Street address
Kirova, 35

City
Drogichin

Phone Number
+375297523029

Payment information

Card number Expiration date CVC

76543 27/20 940

COMPLETE

Рисунок 7.5 – Страница оформления заказа

Что бы каждый раз не вводить данные при оформлении заказа заново, их можно указать в MY ACCOUNT, и они будут предлагаться автоматически при заказе.

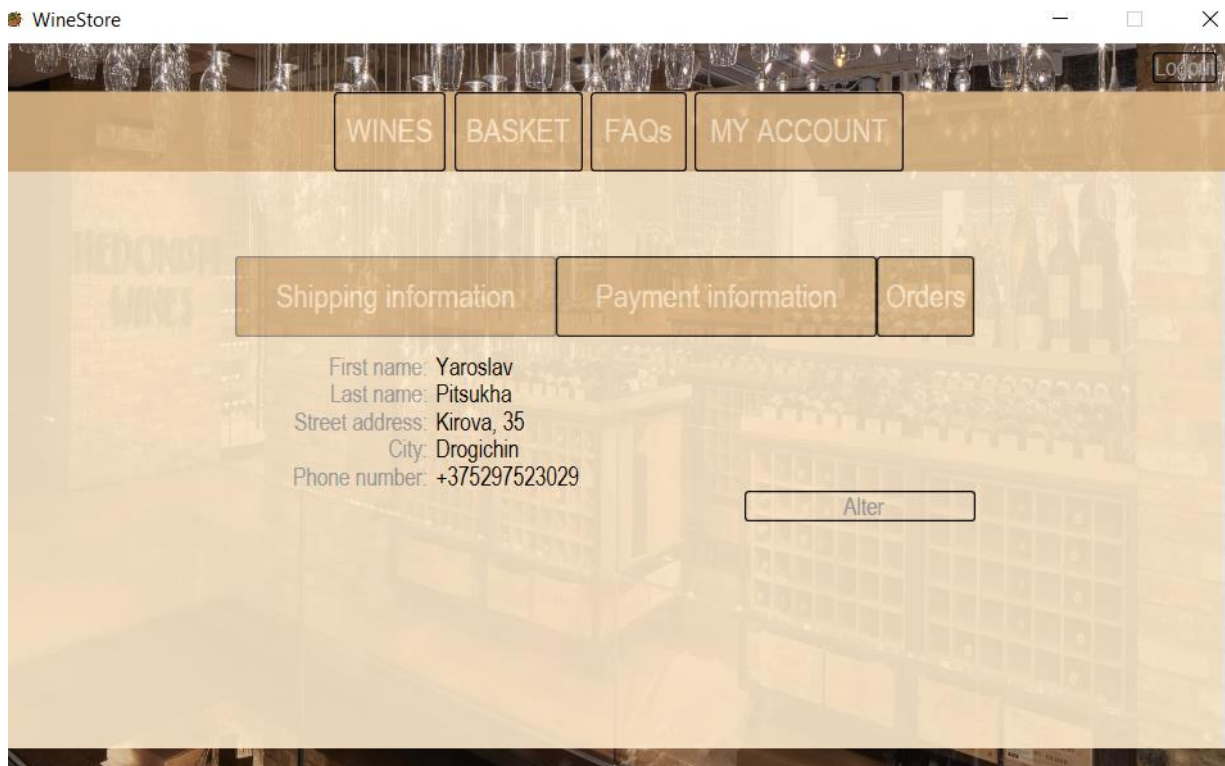


Рисунок 7.6 – Страница с информацией в личном кабинете

Также пользователь может посмотреть информацию о всех своих заказах в MY ACCOUNT на вкладке Orders.

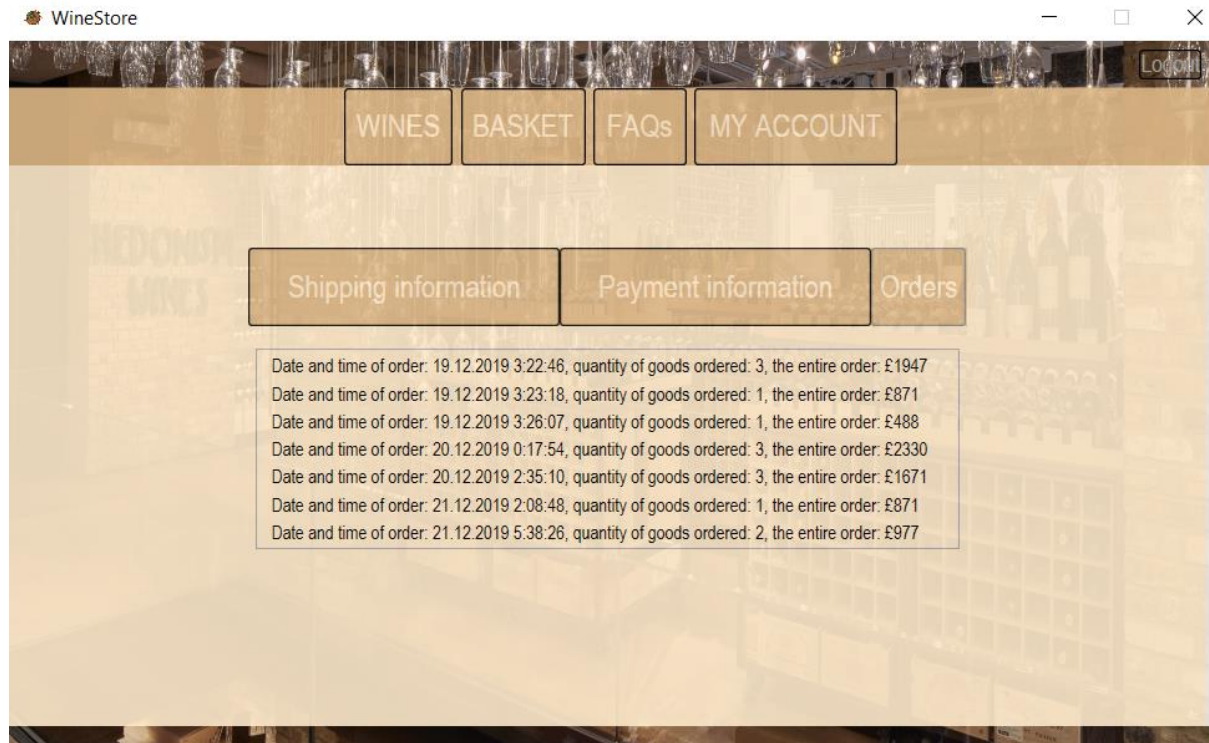


Рисунок 7.7 – Страница со список заказов в личном кабинете

Последняя вкладка, которая доступна пользователю это FAQs на ней он может найти ответы на интересующие его вопросы, связанные с магазином. Для выхода из аккаунта необходимо нажать на кнопку Logout.

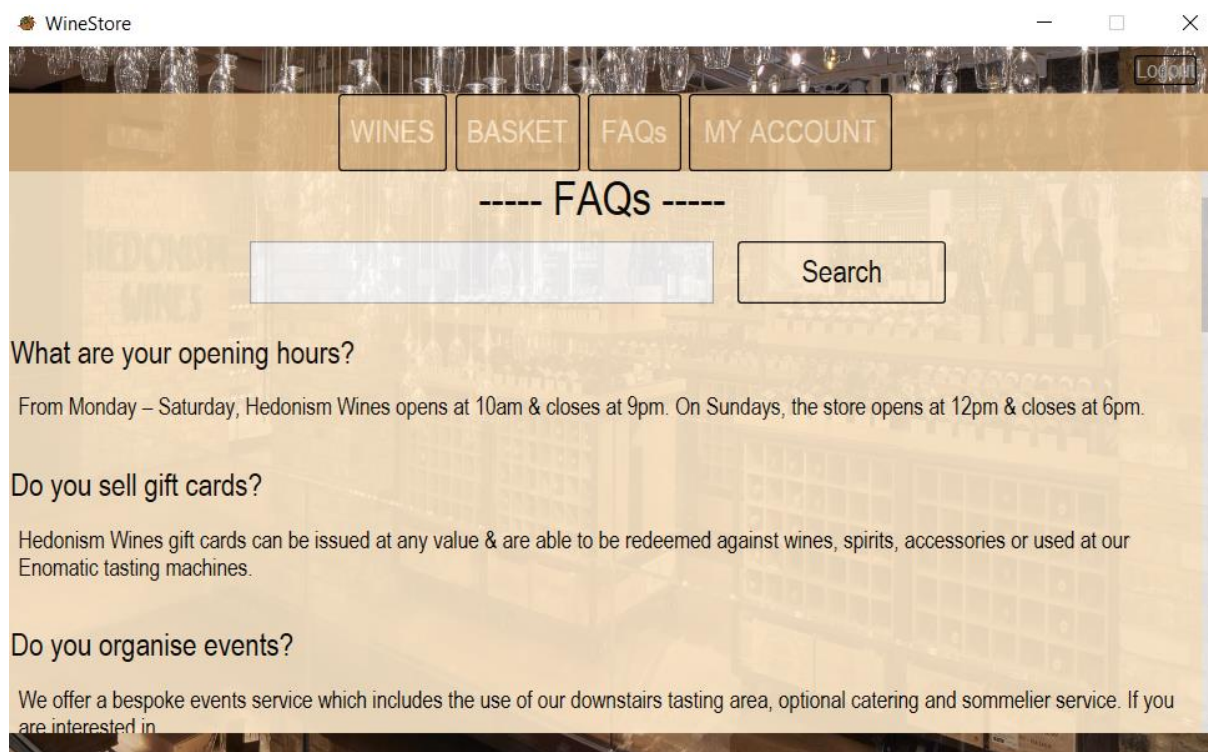


Рисунок 7.8 – Страница раздела вопросов

Если в систему вошёл администратор, то у него отображается отличная от пользовательской главная страница, со своим набором функционала.

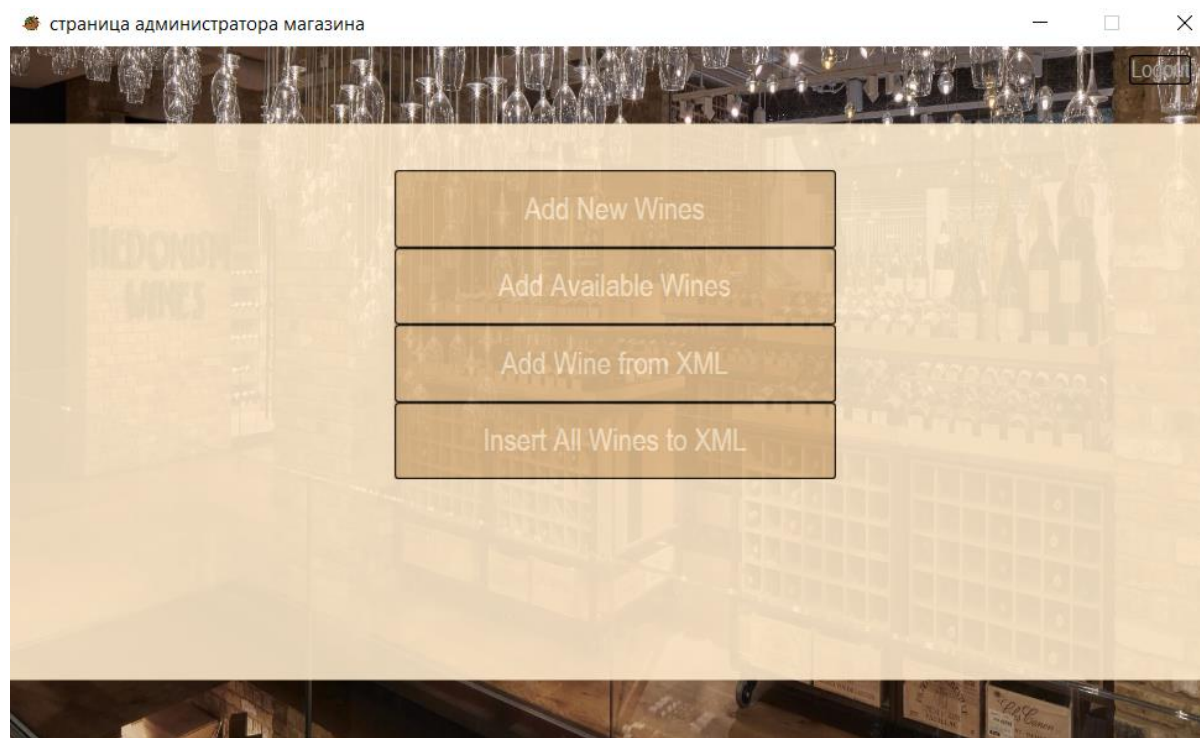


Рисунок 7.9 – Главная страница администратора

Администратор может добавлять данные о новых товарах через указания параметров, либо загрузить готовые товары из XML. Также он может посмотреть список всех имеющихся в каталоге магазина вин в XML формате. Он может редактировать уже имеющиеся товары. Для выхода нужно нажать Logout.

страница администратора магазина

Logout

Product name:

Category:

Type:

Country:

Vintage:

Price:

Volume:

Available:

Path to foto:

Code:

Add to assortment

Рисунок 7.10 – Страница добавления товара

Перечень всех действий пользователя можно увидеть на UML-диаграмме, которая осуществляет визуализацию разрабатываемой системы с разных точек зрения. Варианты использования отражают функциональность ПС с точки зрения получения значимого результата для пользователя. UML-диаграмма вариантов использования представлена в приложении В.

Заключение

Перед началом разработки базы данных был произведен аналитический обзор прототипов приложений подобной тематики и определение функциональных возможностей моей базы данных.

В процессе выполнения курсовой работы была спроектирована база данных для винного магазина. База данных была разработана с помощью системы управления базами данных «Microsoft SQL Server 2012». Помимо этого, была изучена и применена при разработке приложения технология полнотекстового поиска.

При разработке базы данных было созданы следующие объекты:

- 11 таблиц с внешними связями;
- 28 хранимых процедур;
- 11 индексов;
- 2 пользователя базы данных.

Также было разработано приложение, демонстрирующее работу с базой данных. Приложение написано с помощью технологии WPF и языка программирования C#. Связь между базой данных и приложением осуществляется с помощью технологии ADO.NET, которая предоставляет набор классов, через которые мы можем установить подключение к базе данных и производить ряд других операций. Также было написано руководство пользователя для созданного приложения.

В соответствии с полученным результатом, можно сказать, что разработанная база данных функционирует верно, требования технического задания реализованы в полном объеме, в созданном приложении предоставлены возможности базы данных, поэтому цель курсового проекта можно считать достигнутой.

				КП 00.00.ПЗ							
	ФИО	Подпись	Дата								
Разраб.	Пицуха Я. А.			Заключение				Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.									31	1
								БГТУ 71191005, 2019			
Н. контр.	.										
Утверд.											

Список литературных источников

1. Официальный сайт «WineStyle» [Электронный ресурс] – Режим доступа: <https://winestyle.ru/>
2. Официальный сайт «Goodwin» [Электронный ресурс] – Режим доступа: <https://goodwin.wine/>
3. Официальный сайт «Ароматный мир» [Электронный ресурс] – Режим доступа: <https://amwine.ru/>
4. Документация СУБД «Microsoft SQL Server» [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15>
5. Мэтью Макдональд — WPF: Windows Presentation Foundation в .NET 4.0 с примерами на C#
6. Язык C# и .NET Framework [Электронный ресурс] — Режим доступа: http://professorweb.ru/my/csharp/charp_theory/level1/
7. Руководство по ADO.NET и работе с базами данных [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/adonet/>
8. Компонент Full-text search – SQL Server [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/sql/relational-databases/search/full-text-search?view=sql-server-ver15>
9. Официальный сайт «HEDONISM WINES» [Электронный ресурс] – Режим доступа: <https://hedonism.co.uk/>

				<i>КП 00.00.ПЗ</i>			
	ФИО	Подпись	Дата	<i>Список литературных источников</i>			
Разраб.	<i>Пицуха Я.А.</i>						
Провер.	<i>Бондарчик Е.Н.</i>						
Н. контр.	.						
Утверд.				<i>БГТУ 71191005, 2019</i>			
				<i>Лит. Лист Листов</i>			
				<i>32 1</i>			

Приложение А

Приложение Б

Листинг создания таблиц

```
CREATE TABLE "CARD"(  
ID int primary key identity(1,1),  
CARD_NUMBER INT,  
DATE nvarchar(5),  
CVC INT  
);
```

```
CREATE TABLE ABOUT_USER(  
ID int primary key identity(1,1),  
FIRST_NAME nvarchar(50) not null,  
LAST_NAME nvarchar(50) not null,  
STREET_ADDRESS nvarchar(50) not null,  
CITY nvarchar(50) not null,  
PHONE_NUMBER nvarchar(50) not null,  
CARD_INFO int references "CARD"(ID)  
);
```

```
CREATE TABLE "USER"(  
ID int primary key identity(1000000,1),  
LOGIN nvarchar(30) not null unique,  
PASSWORD nvarchar(30) not null,  
ABOUT int references ABOUT_USER(ID),  
IS_ACTIVE int references "SESSION"(ID)  
);
```

```
CREATE TABLE "SESSION"(  
ID int primary key identity(1,1),  
LOGIN nvarchar(30) not null unique,  
ACTIVE bit default(0)  
);
```

```
CREATE TABLE WINE(  
ID int primary key identity(1,1),  
PRODUCT_CODE nvarchar(15) not null unique,  
TITLE nvarchar(200) not null,  
VOLUME int not null,  
PRICE decimal(10,2) not null,  
VINTAGE int not null,  
AVAILABLE int not null,  
TYPE nvarchar(20) not null,  
COUNTRY nvarchar(30) not null,  
COTEGORY nvarchar(30) not null,  
FOTO nvarchar(100) not null
```

);

```
CREATE TABLE BASKET(
ID int primary key identity(1,1),
"USER" int references "USER"(ID),
WINE int references WINE(ID)
);
```

```
CREATE TABLE ORDER_HISTORY(
NUMBER int primary key identity(1,1),
"USER" int references "USER"(ID),
DATE datetime not null,
ProductCount int DEFAULT(1),
Price int NOT NULL,
Info int references ORDER_INFORMATION(ID)
);
```

```
CREATE TABLE "ORDER"(
NUMBER int primary key identity(1,1),
ORDER_NUMBER int references ORDER_HISTORY(NUMBER),
WINE int references WINE(ID)
);
```

```
CREATE TABLE ORDER_INFORMATION(
ID int primary key identity(1,1),
FIRST_NAME nvarchar(50) not null,
LAST_NAME nvarchar(50) not null,
STREET_ADDRESS nvarchar(50) not null,
CITY nvarchar(50) not null,
PHONE_NUMBER nvarchar(50) not null,
CARD_NUMBER INT not null,
DATE nvarchar(5) not null,
CVC INT not null
);
```

```
CREATE TABLE SHOP(
ID int primary key identity(1,1),
ADDRESS nvarchar(100) not null,
TELEPHON_NUMBER nvarchar(20) not null,
Email nvarchar(20) not null,
DESCRIPTION nvarchar(1000),
ALL_WINE int default(0)
);
```

```
CREATE TABLE SHOP_FAQS(
ID int primary key identity(1,1),
question nvarchar(400) NOT NULL,
answer nvarchar(400) NOT NULL,
SHOP_NAME INT references SHOP(ID)
);
```

Листинг создания логинов и пользователей
USE WineStore;

```
CREATE LOGIN ADMINISTRATOR_LOG
WITH PASSWORD = 'admin',
    DEFAULT_DATABASE = [WineStore],
    DEFAULT_LANGUAGE = [english];
```

```
CREATE USER ADMINISTRATOR
FOR LOGIN ADMINISTRATOR_LOG
WITH DEFAULT_SCHEMA = [dbo];
```

```
GO
GRANT EXECUTE ON OBJECT::sp_AddWines TO ADMINISTRATOR;
GRANT EXECUTE ON OBJECT::sp_AlterWines TO ADMINISTRATOR;
GRANT EXECUTE ON OBJECT::sp_ExportWineToXML TO
ADMINISTRATOR;
GRANT EXECUTE ON OBJECT::sp_ImportWineFromXML TO
ADMINISTRATOR;
GO
```

```
go
CREATE LOGIN CLIENT_LOG
WITH PASSWORD = '1111',
    DEFAULT_DATABASE = [WineStore],
    DEFAULT_LANGUAGE = [english];
```

```
go
CREATE USER CLIENT
FOR LOGIN CLIENT_LOG
WITH DEFAULT_SCHEMA = [dbo];
```

```

GO
GRANT EXECUTE ON OBJECT::sp_UserVerification TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_UserLogOut TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_InsertUser TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectALLWines TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectAllWinesSortedPrice TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectAllWinesSortedName TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectAllWinesSortedVintage TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectWines TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_AddWinesToBasket TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectALLWinesToBasket TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_RemoveWineFromBasket TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_CreateOrder TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_InsertOrderInformation TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectALLOrders TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectShippingInformation TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_InsertToAboutUser TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_UpdateShippingInformation TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectPaymentInformation TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_InsertToCard TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_UpdatePaymentInformation TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectAllFaqs TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectFromFaqs TO CLIENT;
GRANT EXECUTE ON OBJECT::sp_SelectShopInfo TO CLIENT;
GO

```

Листинг процедуры sp_AlterWines

```

GO
CREATE PROCEDURE sp_AlterWines
@Code nvarchar(15),
@Price decimal(10,2),
@Available int
AS
BEGIN
if @Price !=0
begin
update WINE
set PRICE = @Price
where PRODUCT_CODE=@Code;
end

if @Available !=0
begin
update WINE
set AVAILABLE = @Available

```

```

where PRODUCT_CODE=@Code;
end
END

```

Процедура импорта данных из XML-файла

```

GO
CREATE PROCEDURE sp_ImportWineFromXML
AS BEGIN
Declare @xml XML

Select @xml = CONVERT(XML,bulkcolumn,2) FROM OPENROWSET(BULK 'D:\3
КУРС\КИ\WineStore\XML\Wine.xml', SINGLE_BLOB) AS X

SET ARITHABORT ON

Insert into Wine
(
PRODUCT_CODE,TITLE,VOLUME,PRICE,VINTAGE,AVAILABLE,TYPE,COUNT
RY,COTEGORY,FOTO
)
Select
      P.value('PRODUCT_CODE[1]','nvarchar(15)') AS PRODUCT_CODE,
      P.value('TITLE[1]','nvarchar(200)') AS TITLE,
      P.value('VOLUME[1]','int') AS VOLUME,
      P.value('PRICE[1]','decimal(10,2)') AS PRICE,
      P.value('VINTAGE[1]','int') AS VINTAGE,
      P.value('AVAILABLE[1]','int') AS AVAILABLE,
      P.value('TYPE[1]','nvarchar(20)') AS TYPE,
      P.value('COUNTRY[1]','nvarchar(30)') AS COUNTRY,
      P.value('COTEGORY[1]','nvarchar(30)') AS COTEGORY,
      P.value('FOTO[1]','nvarchar(100)') AS FOTO
From @xml.nodes('/WineData') PropertyFeed(P)
end;

```

Процедура экспорта данных в формат XML

```

GO
CREATE PROCEDURE sp_ExportWineToXML
@xmlm XML out
as
begin
SET @xmlm = ( Select * FROM Wine
for xml path('Wine'), root('WineData'))
end

```

Процедура sp_SelectWines, процедура применяющая технологию полнотекстового поиска

GO

CREATE PROCEDURE sp_SelectWines

@Title nvarchar(30),

@Category nvarchar(30),

@Type nvarchar(30),

@Country nvarchar(30),

@Vintage int,

@PriceMin int,

@PriceMax int,

@Volume int

AS

BEGIN

SELECT * FROM WINE

WHERE CONTAINS(TITLE, @Title) and COTEGORY = @Category and TYPE = @Type and COUNTRY = @Country and VINTAGE = Vintage and PRICE <= @PriceMax and PRICE >= @PriceMin and VOLUME = @Volume and AVAILABLE > 0;

END

Процедура sp_SelectFromFaqs, процедура применяющая технологию полнотекстового поиска

GO

CREATE PROCEDURE sp_SelectFromFaqs

@Text nvarchar(400)

AS

BEGIN

SELECT question, answer

FROM SHOP_FAQS

WHERE FREETEXT((question, answer), @Text)

END

Приложение В