СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время рынок сервисов и услуг расширяется все более выгодными и интересными предложениями. Крупному и малому бизнесу становится все тяжелее удерживать клиентов из-за постоянно растущей конкуренции. В связи с этим существует необходимость в генерации наиболее креативных и оригинальных идей для удержания и привлечения новых

В сложившейся ситуации постоянной межкорпоративной борьбе за клиентов появилась необходимость в автоматизации бизнес процессов и бизнес логики, что позволит посвящать больше времени на улучшение процесса взаимодействия конечного пользователя с предприятиями, тем самым увеличивая количество постоянных клиентов и увеличивая денежную прибыль компаний.

Разработка информационных систем позволяет ускорять бизнес процессы в несколько десятков раз благодаря тому, что все действия в ней происходят в автоматическом режиме и пользователю не нужно самостоятельно выполнять как прежде множество рутинных операций. Информационная система сделает это за него и в результате вернет запрашиваемые данные, что гораздо удобнее чем традиционные способы работы с предприятиями.

Цель курсового проектирования: разработка информационной системы цветочный салон.

Задачи курсового проектирования:

* провести анализ предметной области;
* разработать техническое задания на программный продукт;
* выполнить анализ задания, выбрать технологию проектирования и разработать проект программного продукта;
* выбрать структуры данных для реализации предметной области программного продукта;
* разработать интерфейс пользователя;
* выбрать язык и среду программирования, наиболее удовлетворяющий проведенной разработки;
* выбрать стратегию и разработать тесты;
* разработать алгоритмы и реализовать их в выборной среде разработки
* выполнить тестирования и отладку программного продукта
* разработать необходимую документацию, указанную в техническом задании

1 АНАЛИТИЧЕСКИЕ АСПЕКТЫ ПО РАЗРАБАТЫВАЕМОЙ ТЕМЕ АСПЕКТА

## Анализ предметной области

Цветочный салон – это место где продаются не только цветы, но и различные принадлежности за уходом комнатных и садовых растений. В день приходят сотни клиентов, поэтому хранить их данные было бы не целесообразно. Подлежат хранению данные тех клиентов, которые сделали заказ через информационную систему. Помимо этого, необходимо хранить данные о самом цветочном салоне.

Информационная система должна включать в себя следующие функции:

* Хранение данных о товарах в магазине, а именно о цветах (которые разбиваются по типам и размерам), о разного рода продукции (горшки, грунт, садовые лопатки, удобрения, пленки, семена растений), а также о количестве имеющихся товаров на складе;
* Хранить данные о сотрудниках цветочного салона (личные данные, такие как фио и данные для авторизации);
* Обеспечивать хранение информации о поставщиках (название фирмы, адрес компании);
* Хранить данные о заказах (данные клиента, заказанном товаре, общей сумме и способе доставки (самовывоз или курьером) и адресом (в случае если выбран самовывоз, то по умолчанию присваивается адрес салона);
* Обеспечивать многоуровневый доступ;
* Обеспечивать выдачу данных по запросу пользователя любого уровня доступа и предоставлять результат в виде таблиц;

Информационной системой будут пользоваться следующие типы пользователей – обычные посетители, которые могут только просматривать текущий ассортимент товаров; зарегистрированные пользователи, которые могут просматривать ассортимент товаров и делать заказы; сотрудники салона, которые могут отслеживать заказы, отправлять их на выдачу; администраторы салона, которые должны будут иметь самый широкий функционал такой как, добавление, обновлении и удаление информации о товарах, просматривать данные о сотрудниках и о клиентах(пользователях), а так же изменять ее;

## Анализ аналогичных приложений

Существует огромное множество аналогичных информационных систем. Рассмотрим один из аналогов - информационная система «Цветы у яблоньки» и ее основной функционал:

* предоставление информации о товарах. Позволяет клиенту посмотреть всю доступную продукцию для приобретения, которая может его заинтересовать и побудить сделать заказ;
* оформление заказов. Доступная при просмотре продукции функция, позволяющая сделать заказ товара в определенном количестве и на определенный адрес;
* статистика. Подсчет заказанных товаров за определенный период времени и отслеживание сезонности – на какое время года приходится большая доля заказов;
* система регистрации и авторизации. Позволяет создать аккаунт в данной информационной системе и использовать его для просмотра продукции, добавления в корзину и оформления заказов;

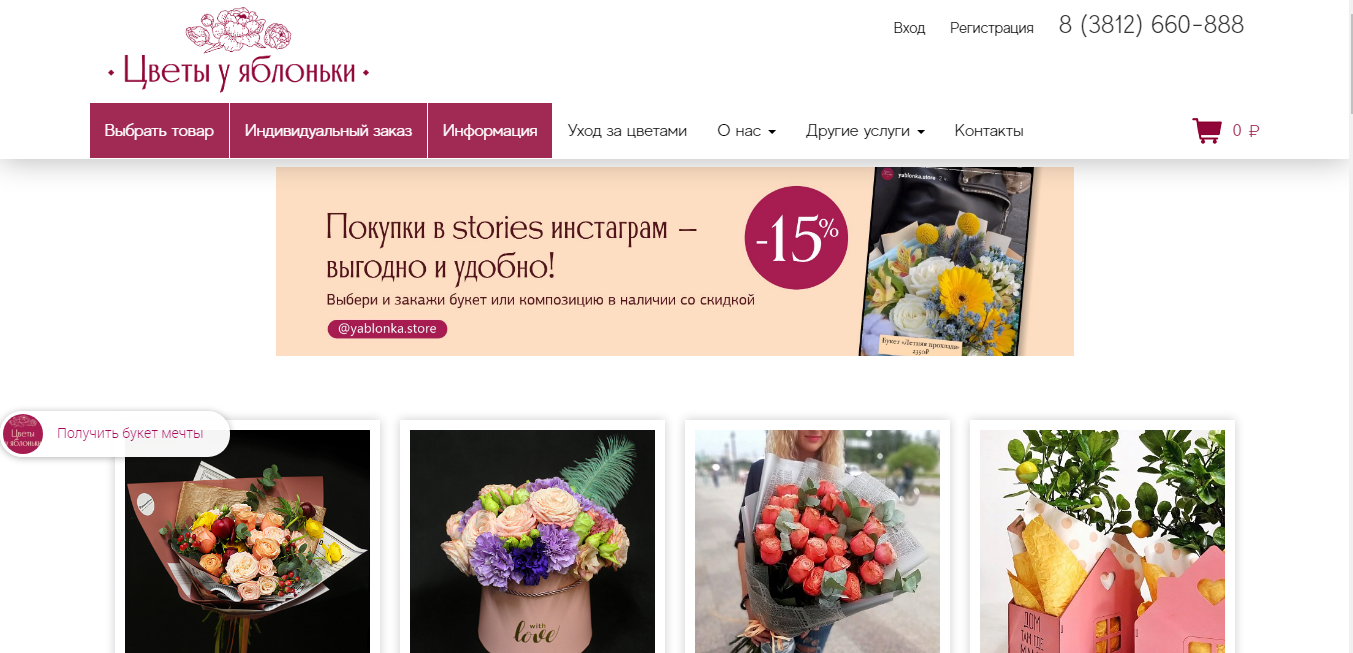


Рисунок 1 – Информационная система «Цветы у яблоньки»

Рассмотрим еще один аналог – информационная система «Букетик55» и ее основной функционал:

* просмотр текущих товаров доступных для заказа и покупки. Клиентам доступно изображение товара, его размеры и стоимость;
* система личного кабинета пользователя. Данная опция позволяет отслеживать свои заказы, их статус, сумму и прочие данные, необходимые для получения оплаченного товара;
* система доставки заказов. Позволяет при оформлении заказа указать адрес доставки, в последствии чего товар будет доставлен курьером;
* возможность создания индивидуального букета. Эта услуга предоставляет клиенту возможность сделать букет по индивидуальному заказу, в котором будут цветы исключительно на усмотрение клиента.

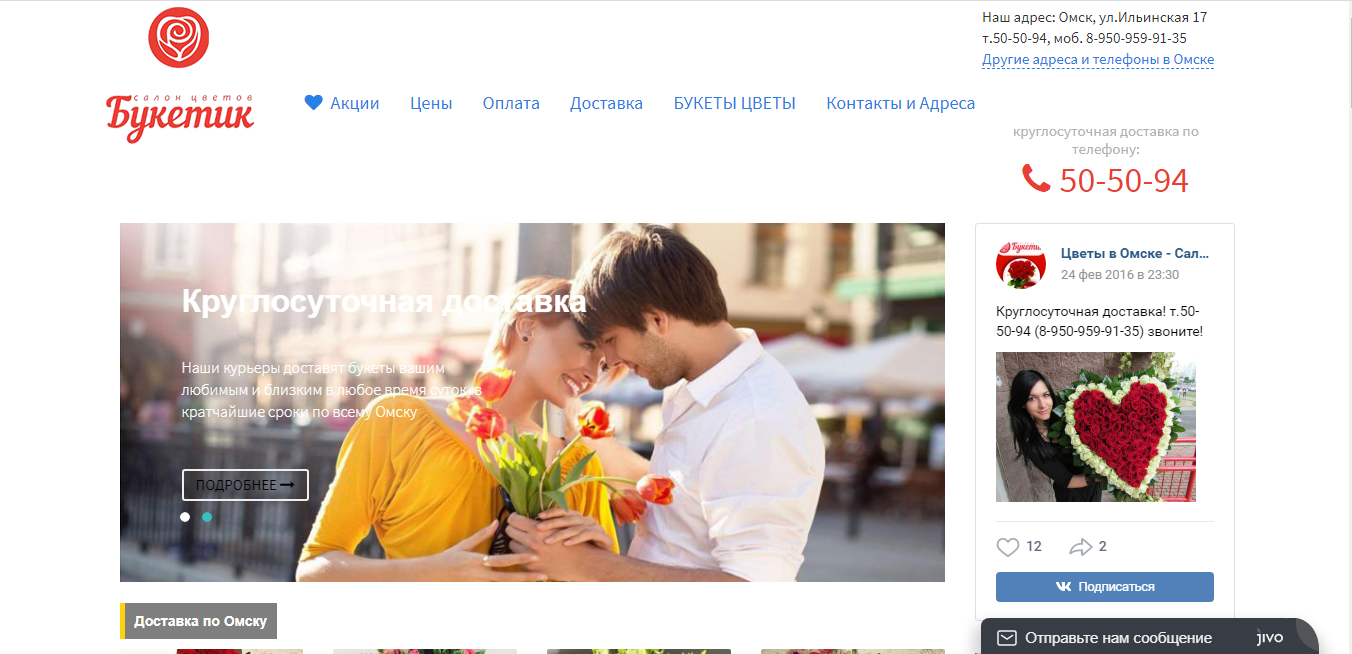


Рисунок 2 – Информационная система «Букетик55»

## Анализ и обоснование выбора программного обеспечения

На рынке представлен огромный выбор программного обеспечения для разработки информационных систем разного уровня сложности и масштабности. Для данного проекта будет использоваться следующее ПО:

* СУБД MS SQL SERVER – система управления базами данных от компании Microsoft, которая великолепно работает в связке с Си подобными языками программирования. Данная СУБД обладает отличной скоростью работы, надежностью и относительной простотой в использовании;
* Язык программирования C# - созданный компанией Microsoft основанный на объектно-ориентированном подходе язык программирования идеально подходит для разработки данной информационной системы в силу своей относительной простоты написания кода, его отладки и поддержки. Немаловажным преимуществом является то, что данный ЯП является компилируемый, что в десятки раз увеличивает скорость работы написанных на нем приложений, нежели интерпретируемых аналогов в виде JavaScript и его Фреймворке Electron или PHP с его Symphony.

2 ПРОЕКТНАЯ ЧАСТЬ РАЗРАБОТАННОГО ПРОЕКТА

## 2.1 Описание структуры данных

Структура данных (англ. data structure) — программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных данных в вычислительной технике. Используется для взаимодействия между формами в приложении.

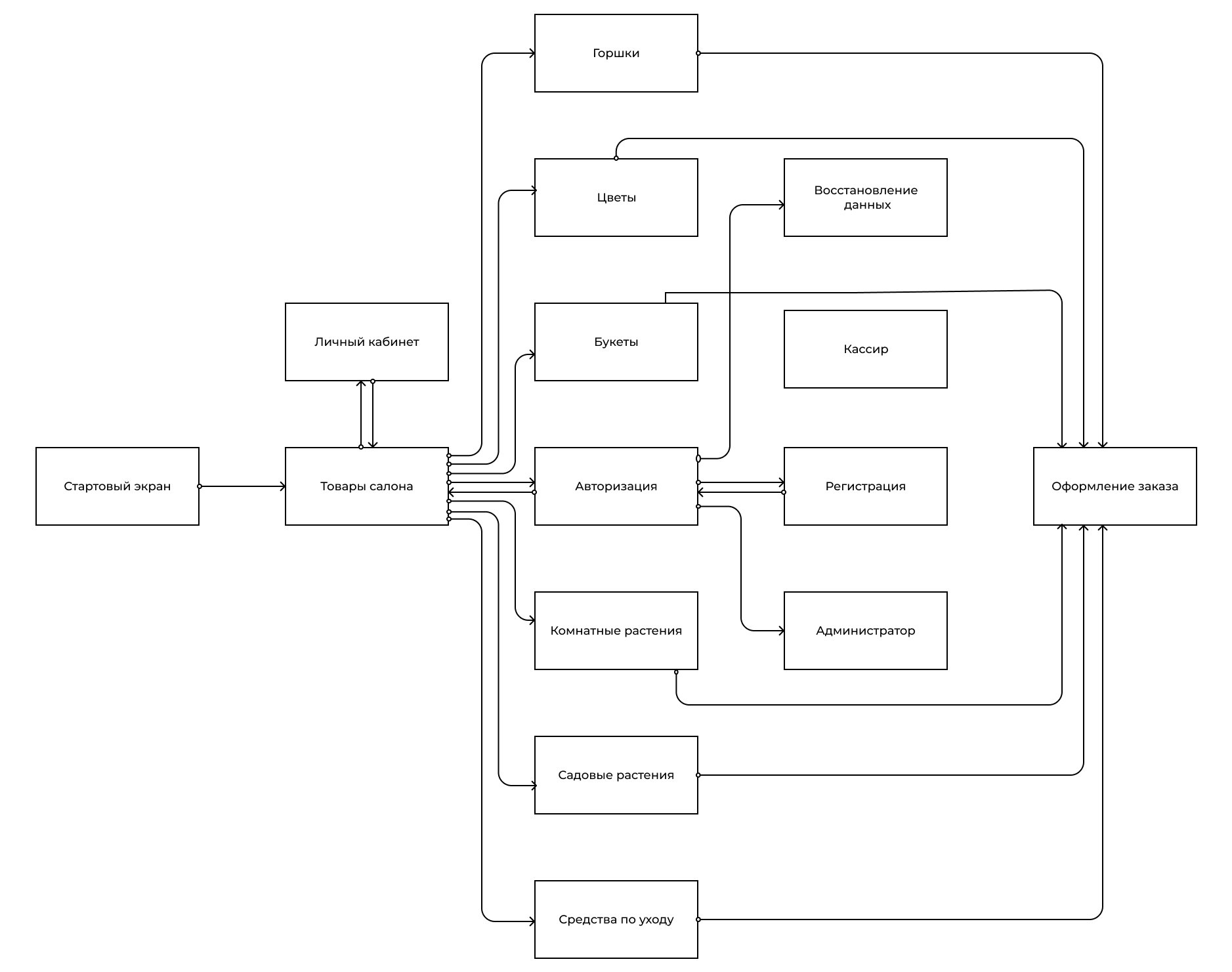


Рисунок 3 – Схема взаимодействия данных

## 2.2 Разработка базы данных

В данном подразделе рассматривается разработка базы данных, а также ее объектов, таких как представления, хранимая процедура и триггеры.

*Создание диаграммы*

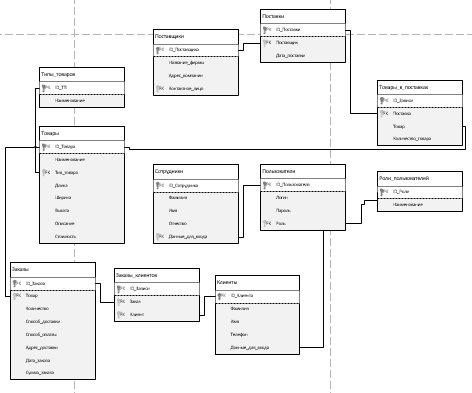


Рисунок 4 – ER-диаграмма базы данных

*Создание таблиц*

Таблица «Пользователи»

Поля таблицы:

* ID\_Пользователя – первичный ключ, позволяющий идентифицировать пользователя. Является целочисленным уникальным значением;
* Логин – поле, хранящее в себе логин пользователя. Строковый тип данных с уникальным значением;
* Пароль – поле, хранящее в себе пароль пользователя для авторизации. Строковый тип данных;
* Роль – поле, отсылающее на таблицу «Роли пользователей» и содержащее в себе роль пользователя. Целочисленный тип данных, внешний ключ;
* Ключ – поле, содержащее в себе секретный десятизначный ключ, генерируемый при регистрации пользователя и позволяющий восстановить с его помощью данные для авторизации.

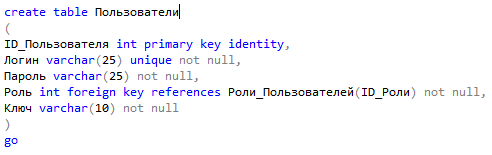


Рисунок 5 – Таблица «Пользователи»

Таблица «Клиенты»

Поля таблицы:

* ID\_Клиента – поле, представляющее собой первичный ключ по которому идентифицируются пользователи. Целочисленный тип данных с уникальным значением;
* Фамилия – поле в которое вносится фамилия клиента при регистрации. Она необходима для оформления и выдачи заказов. Строковый тип данных;
* Имя – поле в которое вносится имя клиента при регистрации. Оно необходимо для оформления и выдачи заказов. Строковый тип данных;
* Телефон – поле в которое вносится телефон при регистрации. Строковый тип данных;
* Данные\_для\_входа – поле, ссылающееся на таблицу «Пользователи» и содержащее в себе данные для авторизации клиента, согласно его пользователю. Целочисленный тип данных, внешний ключ.

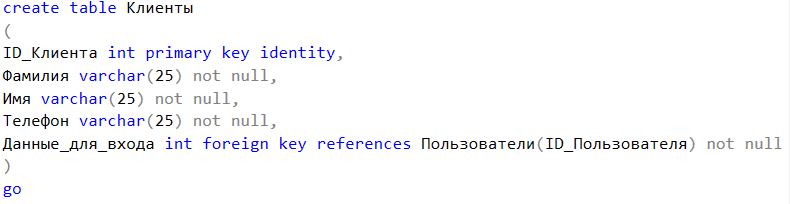


Рисунок 6 – Таблица «Клиенты»

Таблица «Заказы клиентов»

Поля таблицы:

* ID\_Записи – поле, позволяющее идентифицировать записи в таблице. Является первичным ключом с уникальным значением;
* Заказ – поле, являющееся внешним ключом, отсылающим на таблицу «Заказы», которые в свою очередь хранит данные о заказанном товаре, дате заказа, адресе доставки, общей сумме заказа. Целочисленный тип данных;
* Клиент – поле, являющееся внешним ключом, отсылающим на таблицу «Клиенты», которая содержит данные о конкретном клиенте, сделавшим определенный заказ. Целочисленный тип данных;

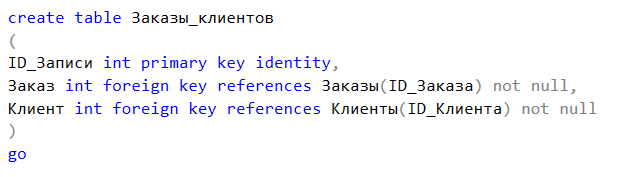


Рисунок 7 – Таблица «Заказы клиентов»

*Создание представлений*

Представление - виртуальная (логическая) таблица, представляющая собой поименованный запрос (синоним к запросу), который будет подставлен как подзапрос при использовании представления.

Представление «GetAllUsersOrders» - позволяет получить заказы всех пользователей, включая данные о номере заказа, заказанном товаре, адресе доставки, фамилии и имени заказчика(клиента), статусе заказа и коде выдачи заказа.

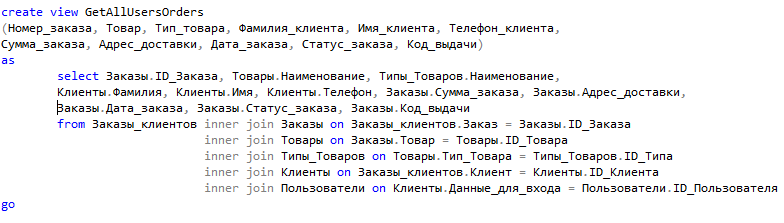


Рисунок 8 – Представление «GetAllUsersOrders»

Представление «GetAllProds» - позволяет получить все данные о товарах цветочного салона, включая тип товара.

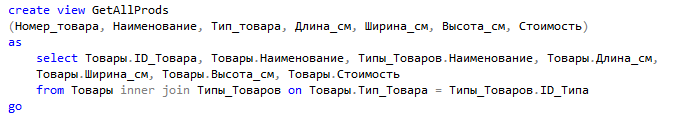


Рисунок 9 – Представление «GetAllProds»

Представление «GetAllEmployees» - позволяет получить все данные о сотрудниках цветочного салона, включая их данные для авторизации в виде логина, пароля и их роли вместе с секретным ключом для восстановления данных.

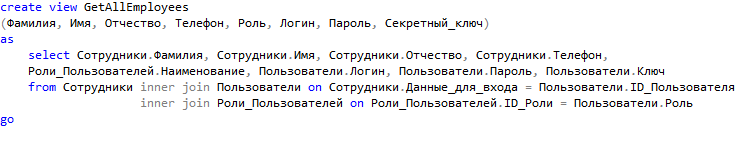


Рисунок 10 – Представление «GetAllEmployees»

*Создание триггеров*

Триггер – хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по модификации данных: добавлением, удалением строки в заданной таблице, или изменением данных в определённом столбце заданной таблицы базы данных.

Триггер «ValidShipmentDateChecker» - позволяет запретить добавление информации о поставке, если дата в поставке больше сегодняшней.

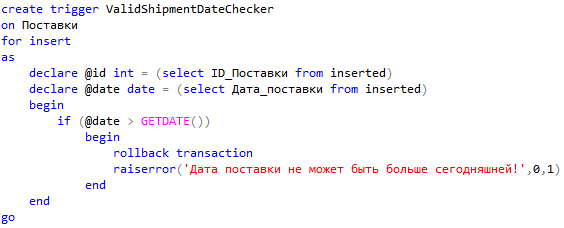


Рисунок 11 – Триггер «ValidShipmentDateChecker»

Триггер «OrderNumberChecker» - позволяет запретить кассиру обновлять статус заказа, которого не существует в базе данных.

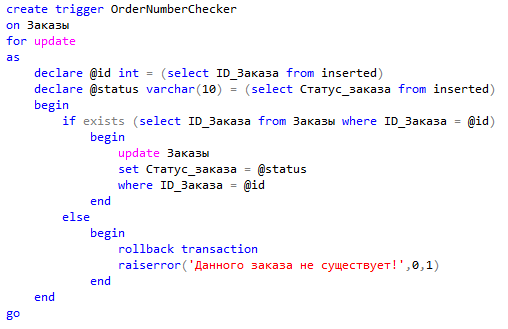


Рисунок 12 – Триггер «OrderNumberChecker»

*Создание хранимой процедуры*

Хранимая процедура – объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере.

Хранимая процедура «GetOrdersByUsername» - позволяет получить данные о заказах конкретного клиента по его логину.

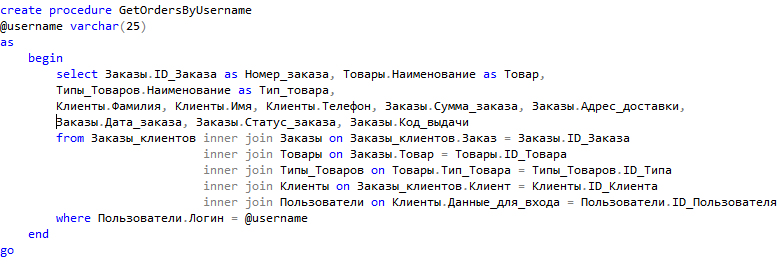


Рисунок 13 – Процедура «GetOrdersByUsername»

## 2.3 Разработка интерфейса программного продукта

Графический интерфейс – необходимая составляющая любой современной информационной системы, которая предназначена для более комфортного и удобного взаимодействия с программой и ее компонентами.

При запуске информационной системы пользователю открывается приветственная форма.



Рисунок 14 – Приветственная форма приложения

После нажатия на кнопку просмотра ассортимента пользователя перенаправляет на форму с продукцией цветочного салона.

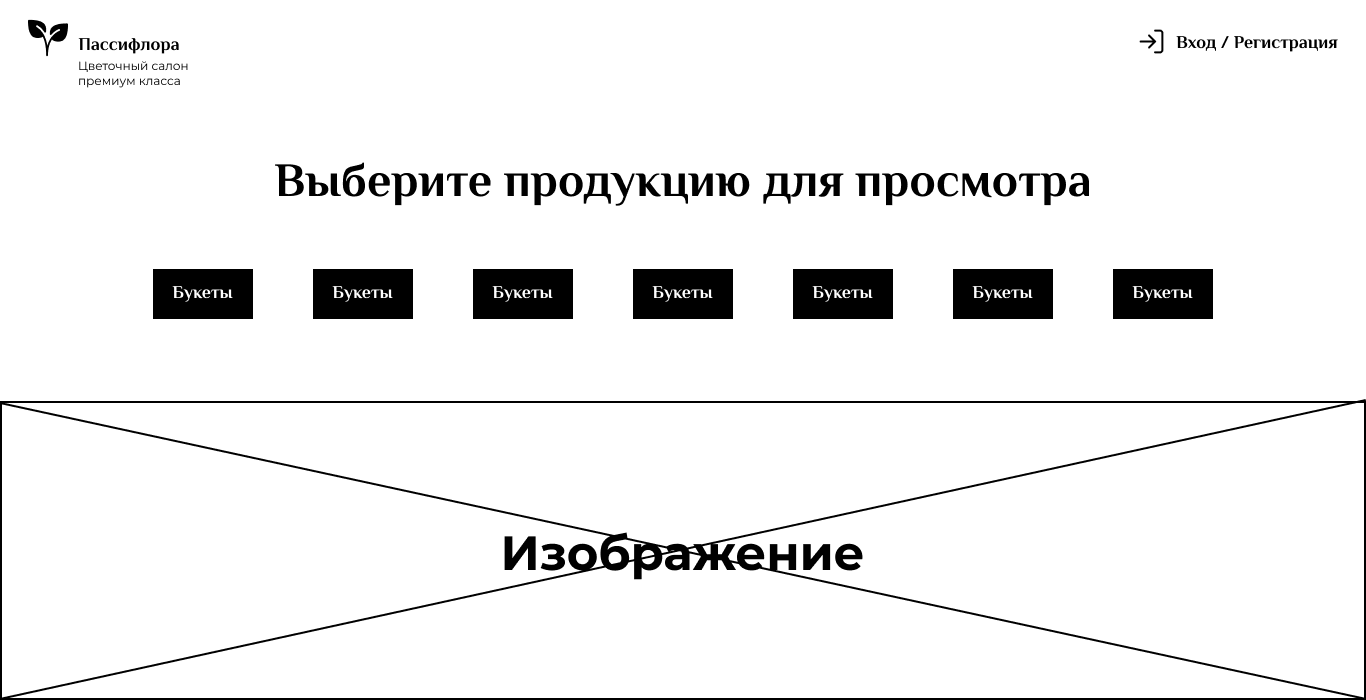


Рисунок 15 – Форма с продукцией цветочного салона

Если пользователь захочет оформить заказ, то ему необходимо авторизоваться или создать учетную запись, если ее нет. Это можно сделать на форме авторизации.



Рисунок 16 – Форма с авторизацией пользователей

## 2.4 Реализация функций программного продукта

## 2.5 Тестирование и отладка программного продукта

Для обработки проблем и ошибок во время эксплуатации программного средства мной было проведено тестирование функционала методом test-case. Этот метод заключается в проверке рутинных операций, которые будут повторяться множество раз. Данное тестирование проверяло работоспособность авторизации в моем программном продукте.

Таблица 1 – Тестирование формы авторизации.

|  |  |
| --- | --- |
| Test Case # | Тест №1 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Проверка формы авторизации |
| Резюме испытания | При успешной авторизации доступ к функционалу в соответствии с учетной записью. |
| Шаги тестирования | 1. Запуск программы  2. Переход на форму с продукцией  3. Переход на форму авторизации  4. Ввод логина  5. Ввод пароля  6. Подтверждение авторизации |
| Данные тестирования | Логин: customer Пароль: customer  Логин: cashier Пароль: cashier  Логин: administrator Пароль: administrator |
| Ожидаемый результат | При вводе Логин: customer Пароль: customer идет запись авторизованного пользователя и перенаправление на форму с товарами цветочного салона.  При вводе Логин: cashier Пароль: cashier происходит запись авторизованного пользователя и перенаправление на панель управления с доступом уровня «Кассир» со своими функциональными возможностями.  При вводе Логин: administrator Пароль: administrator происходит запись авторизованного пользователя и перенаправление на панель управления с уровнем доступа «Администратор», который открывает весь функционал панели управления.  При вводе любых других значений в поля Логин и Пароль будет выведена ошибка о том, что введены некорректные данные. |
| Фактический результат | По работающему приложению |
| Предпосылки | Наличие исполняемого файла |
| Постусловия | При вводе логина и пароля происходит проверка введенных данных на соответствие имеющегося пользователя в базе данных, и если данная проверка успешна, то авторизованного пользователя перенаправляет на форму соответствующего уровня доступа. |
| Статус (Pass/Fail) | По работающему приложению |
| Комментарии | Поле ввода пароля маскирует вводимые символы знаком «•», однако имеется возможность показать пароль, активировав соответствующую функцию на форме авторизации. |

ЗАКЛЮЧЕНИЕ

Список использованных источников

1. Бен-Ган, Ицик Microsoft SQL Server 2014. Основы T-SQL [Текст]: Учебник / Ицик Бен-Ган. - М.: Эксмо, **2019. - 428** c.
2. Гуриков, С. Р. Введение в программирование на языке Visual C# [Текст]: Учебник / Москва: **Высшая школа**, **2017**. - 448 c.
3. Васильев, А.Н. Программирование на C# для начинающих. Особенности языка [Текст]: Учебник / А.Н. Васильев. - М.: ЭКСМО, 2017. - 128 c.
4. Гриффитс, И. Программирование на C# 5.0 [Текст]: Учебник / И. Гриффитс. - М.: ЭКСМО, 2016. - 208 c.
5. Вагнер, Б. Эффективное программирование на C#. 50 способов улучшения кода [Текст]: Учебник / Б. Вагнер. - М.: Вильямс И.Д., 2017. - 224 c.
6. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft. NET Framework 4.5 на языке C# [Текст]: Учебник/ Дж. Рихтер. - СПб.: Питер, 2018. - 351 c.
7. Хейлсберг, А. Язык программирования C#. Классика Computers Science [Текст]: Учебник/ А. Хейлсберг, М. Торгерсен, С. Вилтамут. — СПб.: Питер, 2016. — 784 c.
8. Тейлор, Аллен SQL для чайников [Текст]: Учебник / Аллен Тейлор. - М.: Вильямс, **2018.** - 416 c.
9. Фленов, М.Е. Библия C# (+ CD-ROM) [Текст]: Учебник / М.Е. Фленов. - М.: БХВ-Петербург, **2020. - 176** c.
10. Дунаев, В. В. Базы данных. Язык SQL для студента [Текст]: Учебник / В.В. Дунаев. - М.: БХВ-Петербург, **2017**. - 288 c.
11. Карвин, Билл Программирование баз данных SQL. Типичные ошибки и их устранение [Текст]: Учебник / Билл Карвин. - М.: Рид Групп, **2018**. - 336 c.
12. Молинаро, Энтони SQL. Сборник рецептов [Текст]: Учебник / Энтони Молинаро. - М.: Символ-плюс, **2016**. - 500 c.
13. Проектирование и реализация баз данных Microsoft SQL Server 2000. [Текст]: Учебный курс MCSE (+ CD-ROM). - М.: Русская Редакция, **2017**. - 664 c.
14. Селко, Джо SQL для профессионалов. Программирование [Текст]: Учебник / Джо Селко. - М.: ЛОРИ, **2015**. - 464 c.
15. Форта, Бен Освой самостоятельно SQL за 10 минут [Текст]: Учебник / Бен Форта. - М.: Вильямс, 2015. - 288 c.
16. Гниденко, И. Г. Технология разработки программного обеспечения [Текст]: Учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 235 с.
17. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2 [Текст]: Учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2019. — 501 с.
18. Кубенский, А. А. Функциональное программирование [Текст]: Учебник и практикум для академического бакалавриата / А. А. Кубенский. — М.: Издательство Юрайт, 2019. — 348 с.
19. Кудрявцев, К. Я. Методы оптимизации [Текст]: Учеб. пособие для вузов / К. Я. Кудрявцев, А. М. Прудников. — 2-е изд. — М.: Издательство Юрайт, 2019. — 140 с.
20. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем [Текст]: Учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2019. — 432 с.
21. Стасышин, В. М. Базы данных: технологии доступа [Текст]: Учеб. пособие для СПО / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2018. — 164 с.
22. Трофимов, В. В. Основы алгоритмизации и программирования [Текст]: Учебник для СПО / В. В. Трофимов, Т. А. Павловская ; под ред. В. В. Трофимова. — М.: Издательство Юрайт, 2019. — 137 с.
23. Тухфатуллин, Б. А. Численные методы расчета строительных конструкций. Метод конечных элементов [Текст]: Учеб. пособие для академического бакалавриата / Б. А. Тухфатуллин. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2019. — 157 с.
24. Абрамов, С.А. Математические построения и программирование [Текст]: Учебник / С.А. Абрамов. - М.: Наука, **2016**. - 192 c.
25. Бекишев, Г.А. Элементарное введение в геометрическое программирование [Текст]: Учебник / Г.А. Бекишев, М.И. Кратко. - М.: Наука. Главная редакция физико-математической литературы, **2017**. - 144 c.
26. Мельчук, И.А. Автоматический синтаксический анализ [Текст]: Учебник / И.А. Мельчук. - М.: Редакционно-издательский отдел Сибирского отделения АН СССР, **2018**. - 358 c
27. Албахари Джозеф , Албахари Бен C# 6.0. Карманный справочник [Текст]: Учебник / Вильямс - М., 2015. - **318** c.
28. Культин Н. Microsoft Visual C# в задачах и примерах [Текст]: Учебник / БХВ-Петербург - М., 2015. - 320 c.
29. Подбельский В. В. Язык С#. Базовый курс [Текст]: Учебник / **РГГУ**- Москва, 2015. - 408 c.
30. Пржиялковский, В. В. Введение в Oracle SQL [Текст]: Учебник / В.В. Пржиялковский. - М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, **2016**. - 320 c.