

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра программного обеспечения информационных сетей

К защите допустить:

Заведующая кафедрой ПОИТ
_____ Н.В. Лапицкая

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту
на тему

**ПРОГРАММНОЕ СРЕДСТВО ДЛЯ МАГАЗИНА ДОСТАВКИ ЦВЕТОВ
С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ANGULAR И ASP .NET CORE
(FLOWERS SHOP С ИНТЕГРАЦИЕЙ GOOGLE MAPS, STRIPE,
KEYCLOAK AUTHENTICATION НА БАЗЕ ASP .NET CORE И
ANGULAR)**

БГУИР ДП 1-40 01 01 01 052 ПЗ

Студент

И. В. Ермолович

Руководитель

К. А. Сурков

Консультанты:

от кафедры ПОИТ

К. А. Сурков

по экономической части

Т. А. Рыковская

Нормоконтролер

Г. В. Данилова

Рецензент

Минск 2022

СОДЕРЖАНИЕ

Введение	7
1 Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству	8
1.1 Анализ литературных источников	8
1.2 Обзор аналогов технологий программного средства	10
1.3 Формирование требований к проектируемому программному средству	15
2 Разработка моделей и функциональных требований программного средства для магазина по доставке цветов	18
2.1 Диаграмма вариантов использования	18
2.2 Схема данных программного средства	20
2.3 Инфологическая модель базы данных	22
2.4 Спецификация функциональных требований	24
3 Проектирование программного средства	26
3.1 Разработка архитектуры программного средства	26
3.2 Проектирование модели базы данных	29
3.3 Проектирование и разработка алгоритмов программного средства	31
4 Разработка программных модулей	35
5 Тестирование программного средства	37
6 Руководство пользователя	41
7 Техничко-экономическое обоснование разработки и использования программного средства для магазина доставки цветов с использованием технологий angular и asp .net core	49
7.1 Характеристика разработанного по индивидуальному заказу программного средства	49
7.2 Расчет затрат на разработку и цена программного средства, созданного по индивидуальному заказу	50
7.3 Расчет результата от разработки и использования программного средства, созданного по индивидуальному заказу	53
7.4 Расчет показателей экономической эффективности разработки и использования программного средства	55
Заключение	58
Список использованных источников	59
Приложение А(обязательное). Исходный код программного модуля	61

ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяются следующие определения и сокращения.

Аутентификация – проверка подлинности предъявленного пользователем идентификатора.

Авторизация – предоставление определенному лицу прав на выполнение определенных действий.

Фреймворк – программное обеспечение, облегчающее разработку и объединение различных компонентов большого программного проекта.

СУБД – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Ручное тестирование – тип тестирования, в котором тест кейсы выполняются специалистом по тестированию вручную, без использования инструментов автоматизации.

ПО – программное обеспечение.

ПС – программное средство.

БД – база данных.

ОС – операционная система.

UML – унифицированный язык моделирования.

URL – унифицированный указатель ресурса.

ВВЕДЕНИЕ

По всему миру, независимо от страны, традиций и религии, цветы используют в качестве украшения, а также подарка. Цветами оживляют газоны, дома, залы для торжественных приемов, свадебные мероприятия. Они акцентируют важность события, и придают празднику особое настроение. Для близкого человека они показывают проявление заботы и внимания.

Основной и самый главный недостаток при покупке цветов – их хрупкость. При неаккуратном обращении с ними, их можно испортить, поэтому очень важно сохранить их в целостности и сохранности при переносе из одного места в другое.

Доставка цветов курьером – это отличное решение этой проблемы. Данный способ получения заказа хорош тем, что он экономит время и силы покупателя, а также возлагает ответственность за состояние цветов на службу доставки.

Для того, чтобы максимально автоматизировать и ускорить процесс выбора цветов, оформления заказа, доставки и оплаты, многие магазины реализовали свои веб-приложения, которые включают все эти этапы и позволяют автоматизировать и ускорить их выполнение.

Веб-приложение позволяет не выходя из дома ознакомиться с каталогом, выбрать нужный вариант, оформить заказ и отслеживать его статус. Оно также позволит эффективно рекламировать тот цветочный магазин, которому оно принадлежит, что является важным фактором, так как рынок цветов постоянно растет и находится в условиях жесткой конкуренции. Также благодаря нему магазин может своевременно обновлять актуальную информацию о наличии определенного товара, различных акций, времени работы магазина.

В дипломном проекте будет разработано программное средство для интернет-магазина по доставке цветов. Данное программное средство позволит пользователям просматривать каталог с различными сортировками и фильтрами для удобства, оформлять заказы и отслеживать их статус. Приложение будет иметь удобный пользовательский интерфейс, который также будет адаптивен под мобильные телефоны. Оформление заказа будет сопровождаться удобным выбором адреса доставки, как по вводу вручную с подсказками, так и при помощи карты. Кроме того, в приложении будет реализован чат, через который пользователь сможет уточнить прочую информацию у сотрудника магазина.

1 АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОМУ ПРОГРАММНОМУ СРЕДСТВУ

1.1 Анализ литературных источников

Существует огромное множество интернет-магазинов с различными сферами деятельности. Некоторые оставляют после себя хорошее впечатление от использования, а с некоторых хочется быстрее уйти. Поэтому очень важно при проектировании интернет-магазина в общем случае руководствоваться определенными принципами, которые характерны удобному и качественному интернет-магазину.

Выделяют следующие основные принципы:

- Наличие корзины товаров: пользователь в любой момент времени должен иметь функцию просмотреть содержимое корзины и изменить его по своему усмотрению.

- Удобный каталог товаров: каталог товаров должен содержать различные сортировки и фильтры, пагинацию, а также выглядеть аккуратно и компактно на странице веб-приложения.

- Функция регистрации и авторизации: пользователю необходимо предоставить зарегистрироваться в приложении и авторизоваться в нем. Благодаря этому пользователю в дальнейшем будут предоставляться различные персональные бонусы и акции, а также сократиться количество ввода необходимой информации при заказе. Стоит отметить, что веб-приложение также должно успешно работать с пользователем и без регистрации и авторизации его в системе.

- Выбор способа оплаты: пользователю для удобства нужно предоставить несколько вариантов оплаты. Классическими вариантами являются оплата онлайн при помощи карты, оплата картой курьеру и оплата наличными курьеру.

- Автоматическое заполнение части информации при заказе: часть информации о пользователе система должна заполнять сама, в случае работы с зарегистрированным пользователем.

- Подтверждение заказа: перед тем, как заказ начнут выполнять в самом магазине, пользователь должен подтвердить его по электронной почте.

- Сохранение введенной информации и состояния корзины при перезагрузке страницы [1].

Трудно представить такого человека в современном мире, который бы не владел компьютером или смартфоном, ведь в этих устройствах содержится большой объем данных и информации, необходимые для повседневной жизни. Поэтому веб-приложение должно быть разработано как для компьютеров, так и для мобильных телефонов, чтобы всегда быть доступным для пользователя.

Как компьютеры, так и мобильные телефоны поддерживают самый популярный браузер в мире на данный момент – Google Chrome. Он поддерживается почти всеми устройствами, постоянно обновляется и содержит набор полезных инструментов и функций, таких как:

- Сохранение паролей.
- Синхронизация данных на различных устройствах.
- Работа с вкладками: функция разделения вкладок по группам и сворачивание группами, что облегчает работы с большим количеством открытых вкладок.

– Интеграция с встроенными сервисами Google, такими как Google Translate, Google Disk, Google Documents и прочими.

– Адресная строка: позволяет переводить текст, получать доступ к файлам, высчитывать различные математические выражения и многое другое прямо из адресной строки.

Дополнительно именно браузер Google Chrome поддерживает наибольшее количество функций при разработке веб-приложений именно под него. Учитывая все вышеперечисленное, веб-приложение должно разрабатываться именно по этот браузер [2].

В последние годы особенно популярным для разработки веб-приложений стали технологии ASP .NET Core и Angular. Первая используется для разработки серверной части веб-приложения, вторая для клиентской.

Основными преимуществами ASP .NET Core перед другими технологиями разработки веб-сервера являются:

- Разработка и запуск на ОС Windows, macOS и Linux.
- Внедрение зависимостей встроено.
- Интеграция различных клиентских платформ, в том числе и Angular.
- Модульный конвейер HTTP-запросов упрощен.
- Взаимодействие с базой данных MS SQL при помощи Entity Framework [3].

Вышеперечисленные преимущества существенно упростят и ускорят разработку программного средства. Теперь рассмотрим достоинства Angular по отношению к его конкурентам:

- Процесс написания кода ускорен.
- Код получается максимально структурированным.
- Использование кода повторно.
- Выход обновлений регулярен.
- Тестирование отдельных частей приложения.
- Связывание данных, мгновенно отражающее изменения состояния веб-страницы в пользовательском интерфейсе.
- Производительность.
- Наличие простых шаблонов.
- Индексация в Google [4].

Подводя итог всему вышесказанному, веб-приложение для интернет-магазина будет разработано на ASP .NET Core и Angular, так как именно эти технологии позволят разработать качественное и надежное программное средство.

1.2 Обзор аналогов технологий программного средства

Необходимо проанализировать схожие приложения, для составления общей структурной модели и планирования уникальных функций, которые будут выделять данное программное средство на фоне остальных.

1.2.1 Веб-приложение «Розы.бел»

Веб-приложение «Розы.бел» – один из самых популярных вариантов, предлагаемых по результатам поисковой строки Google. Данное приложение имеет большой список функций, предоставляемых пользователю:

- Просматривать каталог.
- Выбирать предпочитаемую сортировку.
- Включать фильтры.
- Выполнять поиск по каталогу.
- Редактировать корзину.
- Выбирать способ доставки и оплаты при оформлении заказа.

Стоит отметить, что вся расположенная на странице информация легко распознается при первом просмотре и не вызывает затруднений с ее поиском.

Однако, приложение имеет и ряд серьезных недостатков, основные из которых:

- Отсутствие онлайн-чата, что ведет к невозможности уточнить информацию без телефонного звонка.
- Интерфейс значительно устарел.
- Выбор адреса доставки по карте не реализован.

Данные недостатки не являются критичными, но могут вызвать затруднения у некоторых пользователей.

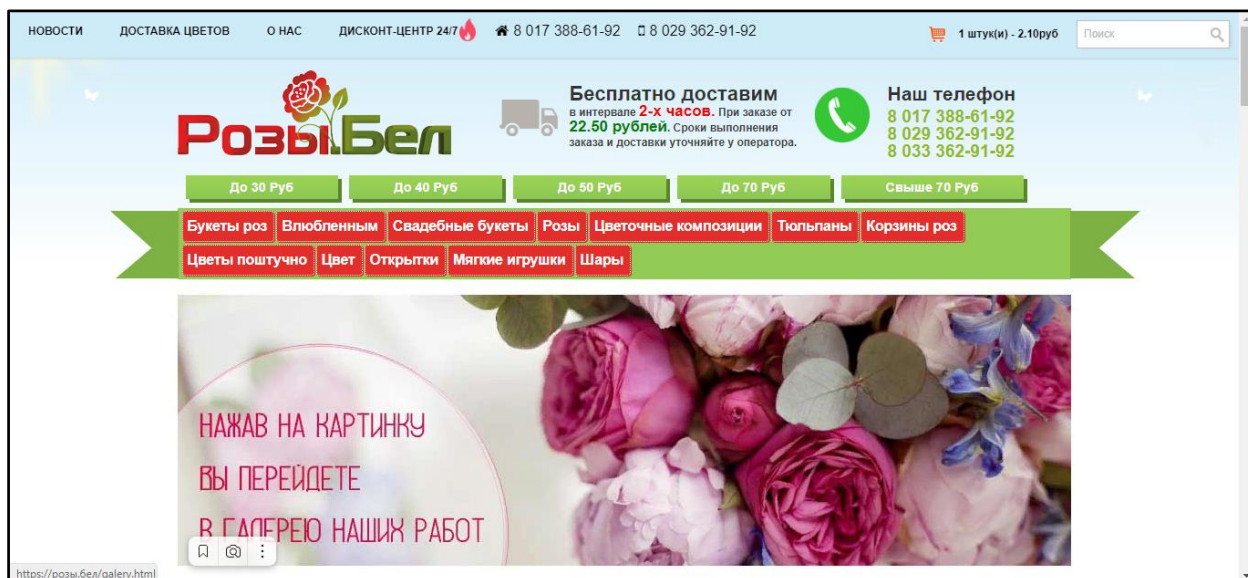


Рисунок 1.1 – Главная страница в приложении «Розы.бел»

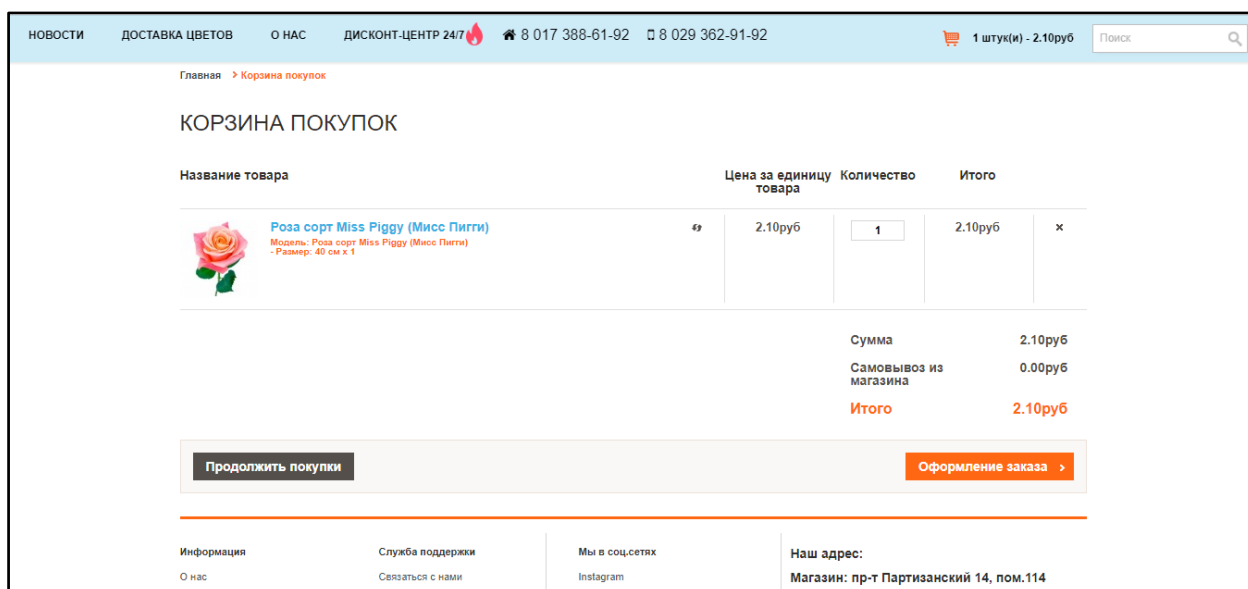


Рисунок 1.2 – Страница корзины в приложении «Розы.бел»

Оформление заказа

Личные данные Заказчика
(для связи с Вами и уточнения заказа после оформления)

* Имя, Фамилия:

* E-Mail:

* Телефон:
Пример: +375-33-999-99-99

Адрес доставки

* Адрес:

* Страна:

* Регион / Область:

Способ доставки

Выберите удобный способ доставки для этого заказа:

Самовывоз

☒ Самовывоз из магазина 0.00руб

Фиксированная стоимость доставки

☐ Фиксированная стоимость доставки 5.00руб

Способ оплаты

Выберите способ оплаты для этого заказа:

☒ NutkiGrosh (ЕРИП) / Дебетовая карта

☐ Оплата при доставке

Корзина покупок

Фото:	Наименование товара:	Количество:	Цена:	Итого:
	Роза сорт Miss Piggy (Мисс Пигги)	1	2.10руб	2.10руб

Рисунок 1.3 – Страница оформления заказа в приложении «Розы.бел»

1.2.2 Веб-приложение «Daflor.by»

Данное веб-приложение также создано с целью предоставить пользователю удаленно произвести покупку цветов. Оно является одним из самых популярных в белорусском сегменте интернета.

Указанное веб-приложение имеет следующий функционал:

- Наличие страницы с акциями.
- Просмотр каталога.
- Выбор сортировки.
- Включение фильтров.
- Поиск по каталогу.
- Корзина.
- Выбор способа оплаты и доставки.
- Связь с сотрудниками через Telegram, Viber.

Одной из отличительных черт от предыдущего рассматриваемого аналога является современный и улучшенный пользовательский интерфейс. Сочетание выбранных цветов и общая стилистика страниц выглядит намного лучше предыдущего примера.

Рассматриваемый пример не избежал и ряда недостатков:

- Отсутствие функции регистрации и авторизации с дальнейшим ускорением процесса оформления заказа.
- Уточнение наличия товара без связи с сотрудниками магазина невозможно.

Исходя из вышеперечисленных фактов, данное приложение исправило некоторые недостатки предыдущего и привнесла некоторый дополнительный функционал. Однако даже данный аналог требует улучшения.

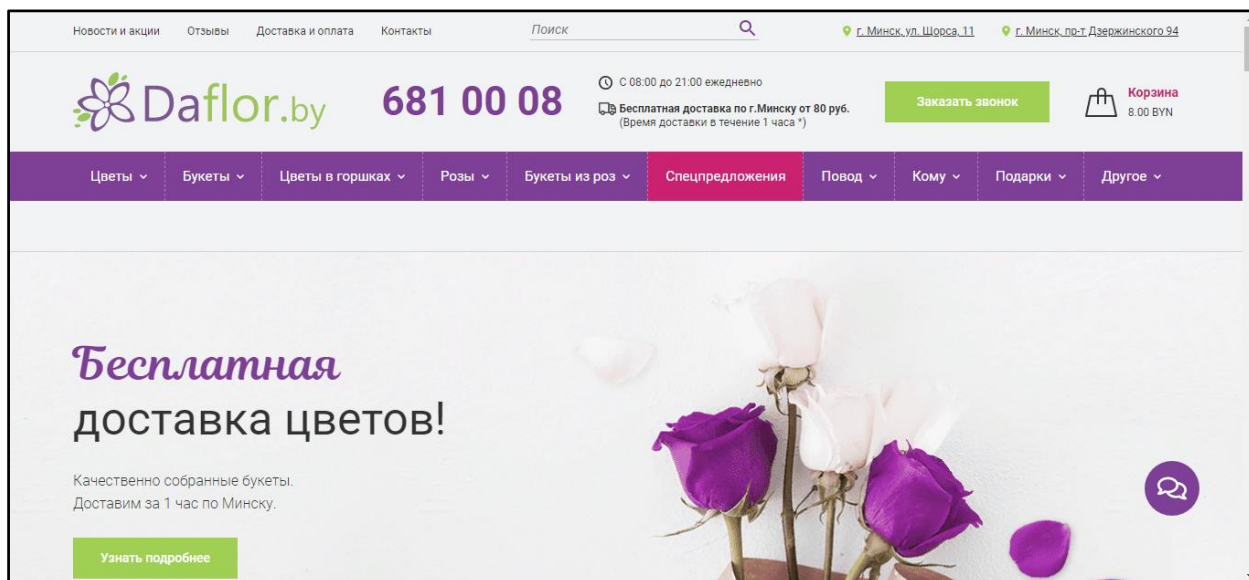


Рисунок 1.4 – Главная страница приложения «Daflor.by»

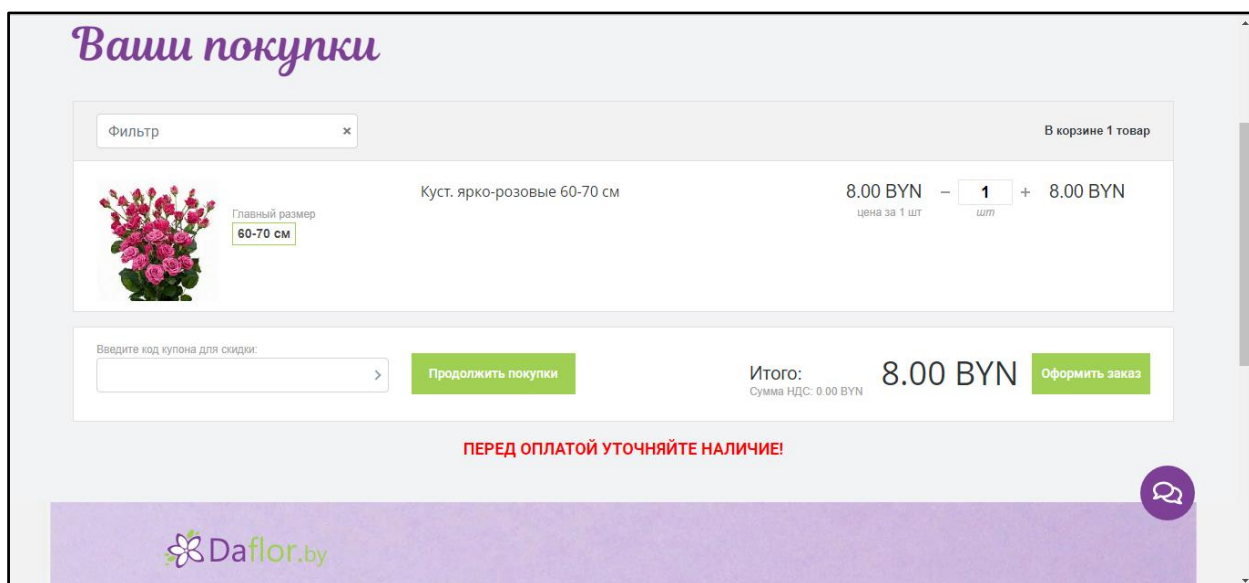


Рисунок 1.5 – Страница корзины приложения «Daflor.by»

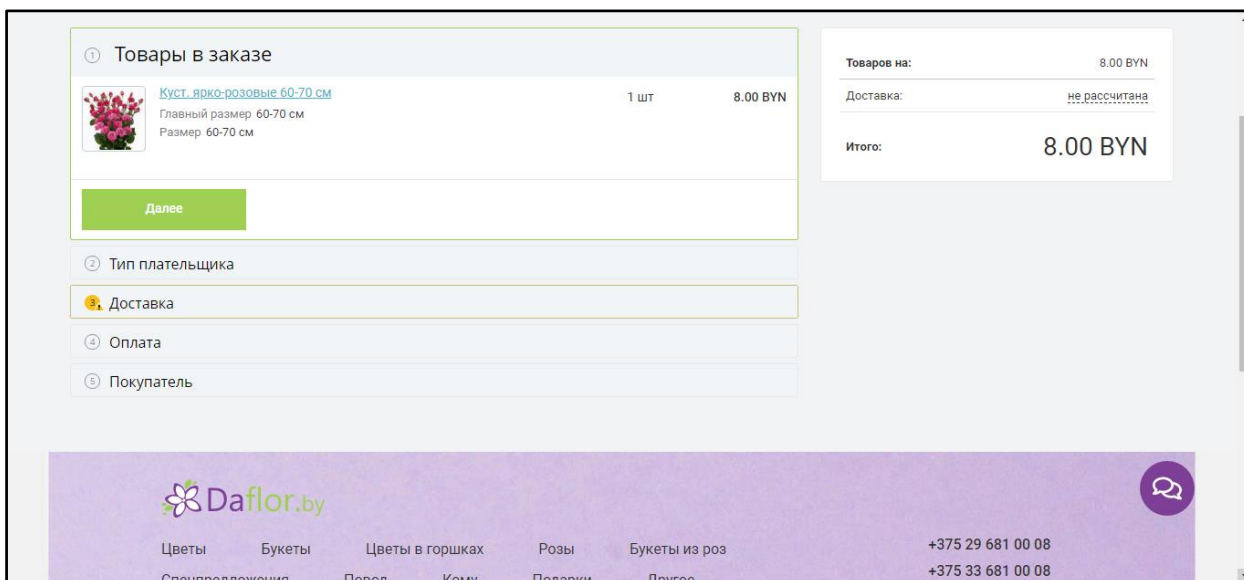


Рисунок 1.6 – Страница оформления заказа приложения «Daflor.by»

1.2.3 Веб-приложение «Podarok.by»

Последним рассматриваемым примером будет программное средство «Podarok.by». Оно предназначено не только для покупки цветов, а также других видов товаров. Данный интернет-магазин признан лучшим в Республике Беларусь по результатам премии E-Commerce 2021.

Данное приложение содержит следующие функции:

- Просмотр акций.
- Просмотр каталога, включающий сортировки, фильтры, поиск, пагинацию, товар включает в себя информацию о наличии.
- Корзина с функцией редактирования.
- Выбор способа оплаты и доставки.
- Связь с сотрудниками как через социальные сети, так и через онлайн-чат.
- Функция регистрации и авторизации с сохранением личных данных.

Серьезных недостатков у данного аналога не было замечено, основные функции для пользователей реализованы максимально удобно. Стоит также отметить, что данный интернет-магазин позаботился и о мобильных приложениях под операционные системы Android и IOS.

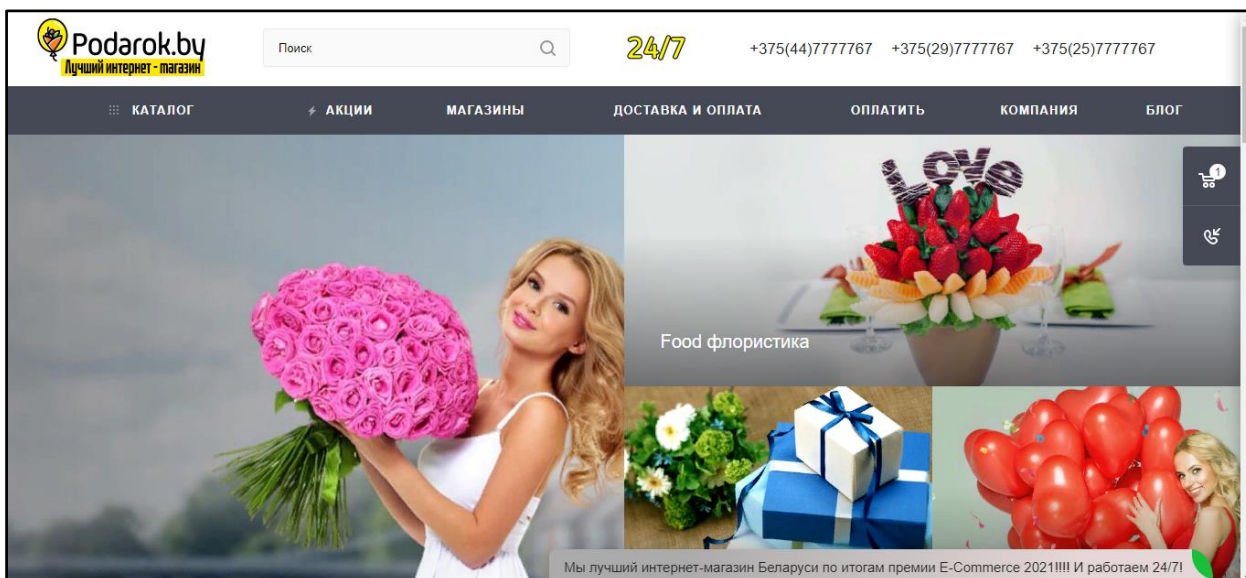


Рисунок 1.7 – Главная страница приложения «Podarok.by»

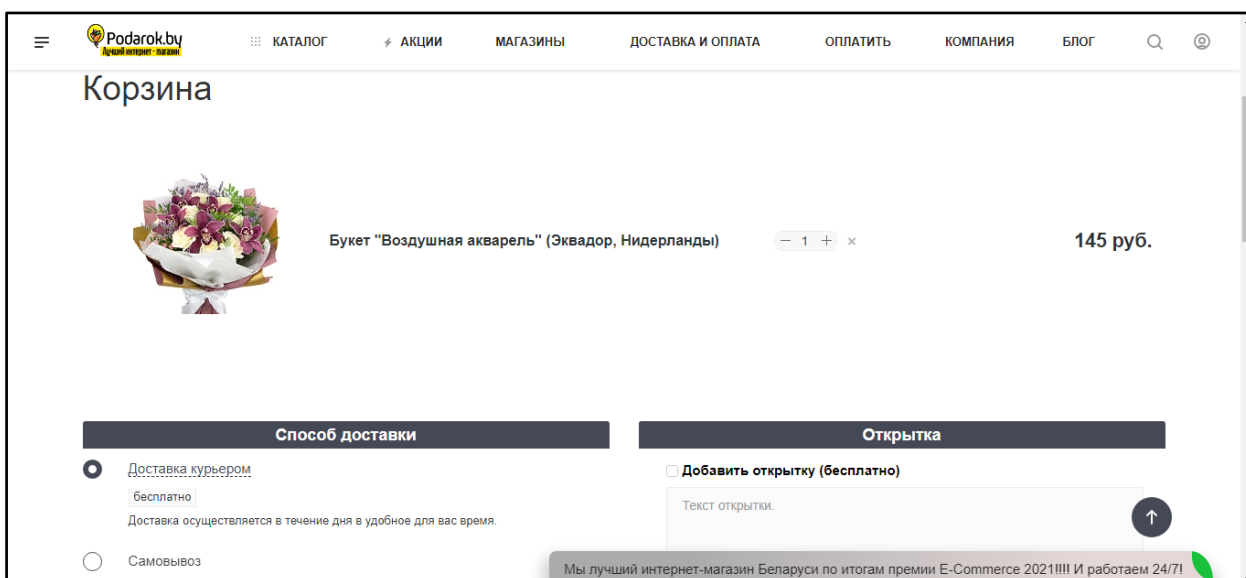


Рисунок 1.8 – Страница корзины и оформления заказа приложения «Podarok.by»

1.3 Формирование требований к проектируемому программному средству

Программное средство для интернет-магазина по доставке цветов, которое необходимо разработать, должно предоставлять пользователю все функции, необходимые для удаленной покупки цветов.

Приложение должно обеспечивать выполнение следующего списка функций:

- а) Просмотр каталога с реализацией функций:
 - 1) Фильтрации.

- 2) Сортировки.
 - 3) Поиска.
 - 4) Добавления товара в корзину.
 - 5) Перехода на страницу товара.
- б) Редактирование корзины.
- в) Оформление заказа, включающее:
- 1) Ввод адреса доставки.
 - 2) Выбор способа оплаты.
 - 3) Ввод прочей контактной информации.
- г) Онлайн-оплата заказа картой.
- д) Переход на страницу:
- 1) Новостей магазина.
 - 2) Контактной информации магазина.
 - 3) Списка заказов со статусами заказа.

Создание дополнительных функций считается возможным в случае привлечения к проекту дополнительных долгосрочных инвестиций.

Разрабатываемое программное средство должно соответствовать следующим требованиям, перечень которых был сформирован по итогам анализа литературных источников и существующих аналогов:

- Реализация удобного и адаптивного интерфейса программы.
- Наличие чата для связи с сотрудниками магазина.
- Функция регистрации и авторизации нового пользователя.
- Страница личного кабинета пользователя.
- Выбор адреса доставки по точке на карте.

Входные данные для веб-приложения должны быть представлены пользователем в качестве его действий и ввода текстовой и иной информации в приложении.

Выходные данные должны быть представлены посредством отображения информации при помощи различных элементов реализованного и доступного пользовательского интерфейса. Непосредственно результатом работы приложения является реакция приложения на различные действия пользователя.

Для надежного функционирования программного средства от заказчика, а также команды поддержки приложения необходимо выполнение следующего комплекса административных и организационно-технических мероприятий:

Выполнение рекомендаций Министерства труда и социальной защиты РБ, изложенных в Постановлении от 23 марта 2011 г. «Об утверждении Норм времени на работы по обслуживанию персональных электронно-вычислительных машин, организационной техники и офисного оборудования».

Выполнение требований ГОСТ 31078-2002 «Защита информации. Испытания программных средств на наличие компьютерных вирусов».

Организация бесперебойного питания.

Время на восстановление работоспособности системы после отказа, вызванного фатальным сбоем операционной системы или техническими неисправностями используемых аппаратных средств, не должно превышать время, необходимое для устранения текущих неисправностей или переустановку операционной системы с последующим перезапуском программного средства.

Время на восстановление работоспособности системы после отказа, вызванного не фатальным сбоем операционной системы, не должно превышать время, необходимое для перезагрузки операционной системы и запуска программного средства.

ЭВМ, на которой должна функционировать серверная часть программного средства, должна обладать следующими минимальными характеристиками:

- Процессор Intel Core i7.
- Жесткий диск объемом 1 Тб.
- Оперативная память 32 Гб.
- Сетевая карта Ethernet 200 МБит/с.

Также для функционирования серверной части необходим установленный ASP .NET Core сервер и база данных MS SQL 2019.

Клиентская часть программного средства должна функционировать на ЭВМ со следующими минимальными характеристиками:

- Процессор Intel Core с тактовой частотой 2 ГГц и более.
- Оперативная память 4 Гб и более.
- Функция выхода в сеть Интернет.

Для корректной работы программного средства необходим один из следующих браузеров с соответствующей минимальной версией:

- Google Chrome 70.
- Opera 58.
- Mozilla Firefox 66.
- Apple Safari 12.0.
- Microsoft Edge 44.

2 РАЗРАБОТКА МОДЕЛЕЙ И ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ ПРОГРАММНОГО СРЕДСТВА ДЛЯ МАГАЗИНА ПО ДОСТАВКЕ ЦВЕТОВ

С целью описания функций программного средства была выбрана диаграмма вариантов использования в спецификации языка графического описания UML.

UML – это унифицированный язык моделирования, система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем [5].

Основным преимуществом проектирования с использованием языка UML является возможность посмотреть на задачу с разных точек зрения. Другим специалистам также легче понять суть задачи.

Недостатками проектирования с использованием языка графического описания UML является трата времени и необходимость изучения различных диаграмм и их нотаций.

2.1 Диаграмма вариантов использования

Функциональная модель представляется в виде диаграммы вариантов использования, которая отражает отношения актеров и прецедентов.

Данная диаграмма позволяет описать систему на концептуальном уровне. На диаграмме актером является множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Актером может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности. Прецедентом является спецификация последовательностей действий, которая может осуществлять система, подсистема или класс, взаимодействуя с внешними действующими лицами (актерами) [5].

Исходя из выдвинутых требований, проектируемое программное средство предполагает поддержку системы ролей, т.е. разделение пользователей на различные роли, а, следовательно, и выдача различных прав нескольким категориям пользователей: работнику магазина, гостю и авторизованному пользователю.

Регистрация, аутентификация и авторизация – функции, которые доступны для роли «Гость» (пользователь, не зарегистрированный в системе). В приложении будет использоваться Keycloak Authentication, которая позволяет зарегистрироваться и авторизоваться пользователю также через социальные сети. После регистрации или авторизации пользователь получает роль, включающую различные функции.

На представленной диаграмме вариантов использования присутствует все вышеописанные роли. Диаграмма представлена на рисунке 2.1

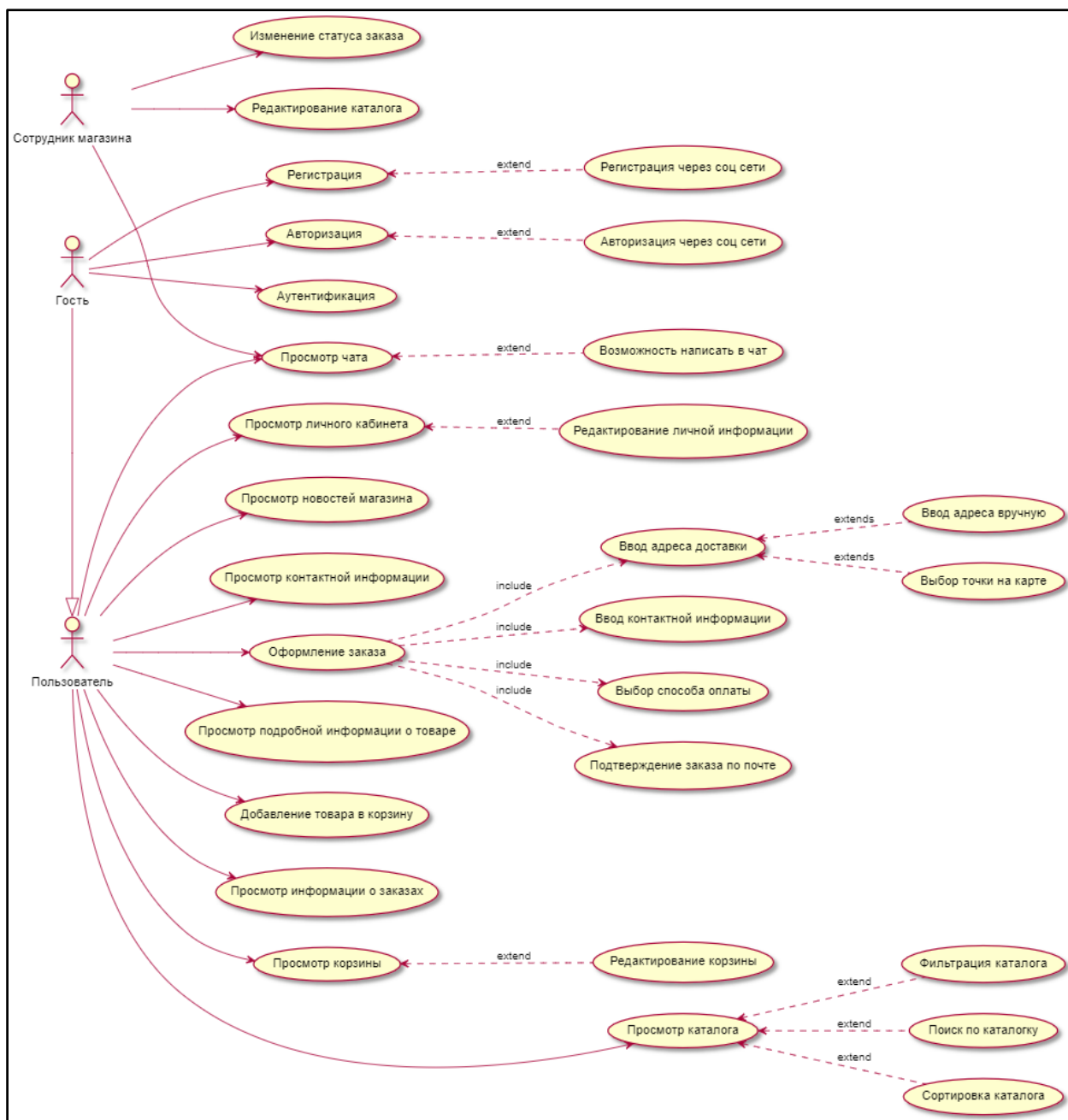


Рисунок 2.1 – Диаграмма вариантов использования программного средств

Рассмотрим детально представленные на рисунке прецеденты.

Роль пользователя является ключевой в данной системе. Она имеет следующие функции:

- Работа с каталогом. Сюда входят функции поиска, сортировки и фильтрации по нему.

– Просмотр личного кабинета. Данная функция позволяет просмотреть личную информацию о пользователе. Сюда входит и функция изменения этих личных данных пользователем при необходимости.

– Добавление товара в корзину. Необходимо для сбора информации по покупаемому товару и его количеству. Просмотр контактной информации. Данная функция позволяет пользователю узнать номер, адрес и прочую информацию о магазине.

– Оформление заказа. В данной функции пользователь вводит всю необходимую для сотрудников магазина информацию о заказе. Сюда входит ввод адреса доставки, путем ручного ввода или выбора точки на карте, выбор способа оплаты, ввод прочей информации, а также подтверждение заказа по электронной почте.

– Функция просмотра информации о заказах. Позволяет пользователю смотреть список всех его заказов, как выполненных, так и выполняемых в данный момент и детальной информации о них.

– Отображение подробной информации о товаре. Необходимо для того, чтобы пользователь мог узнать больше информации о покупаемом товаре, чем имеется в каталоге.

– Наличие новостей магазина. Благодаря данной функции пользователь сможет узнать о новых поступлениях в магазин, различных акциях.

Ответственным за актуальностью содержащийся в приложении информации является сотрудник магазина. Рассмотрим его основные функции:

– Редактирование каталога. Включает добавление товара, удаление товара и редактирование информации о товаре.

– Изменение статуса заказа. Сотрудник магазина оперативно меняет статус заказа, чтобы обеспечить пользователям постоянно знать состояние их заказа и на каком этапе выполнения они сейчас находятся.

– Просмотр чата. Для связи пользователей с сотрудником магазина в программном средстве будет предусмотрен чат. Сообщения будут доставляться мгновенно в реальном времени.

Последней ролью в приложении будет роль гостя. Она состоит из тех же функций, что и роль зарегистрированного пользователя. Добавляется функция зарегистрироваться, аутентифицироваться и авторизоваться.

Таким образом, были описаны все существующие роли. Каждый актер является важной частью данной модели и выполняет свойственные ему функции.

2.2 Схема данных программного средства

На основании составленной функциональной модели программного средства, была разработана схема данных, показывающая путь данных и этапы их обработки.

Схема данных программного средства представлена на рисунке 2.2.

На представленной диаграмме описывается краткая работа с документами. Этапы обработки данных:

1) Начинается схема с символа данных, который отображает список цветов.

2) Далее следует блок процесса, который отображает каталог цветов. От пользователя на данном этапе требуется ввод текста для поиска цветов.

3) После изображен блок процесса, который осуществляет поиск цветов.

4) В результате поиска пользователь получает необходимую информацию о сущности, которая отображается на схеме данных при помощи символа документ.

5) Далее отображен блок процесса получения информации о корзине и информация о сущности, представляющую корзину.

6) Далее у пользователя появляется выбор операции над информацией о корзине и цветке. Пользователь добавить или удалить цветок в корзину, а также начать процесс оформления заказа.

7) Если пользователь выбирает добавление или удаление цветка в корзину, то затем следует блок процесса отображения каталога с цветами.

8) Далее следует символ данных, представляющий из себя список цветов.

7) Если пользователь выбирает переход на страницу оформления заказа, то дальше требуется ввод необходимой информации для заказа.

8) Символом процесса дальше показана функция открытия страницы заказов.

9) Последним символом является символ данных, представляющий из себя список заказов.

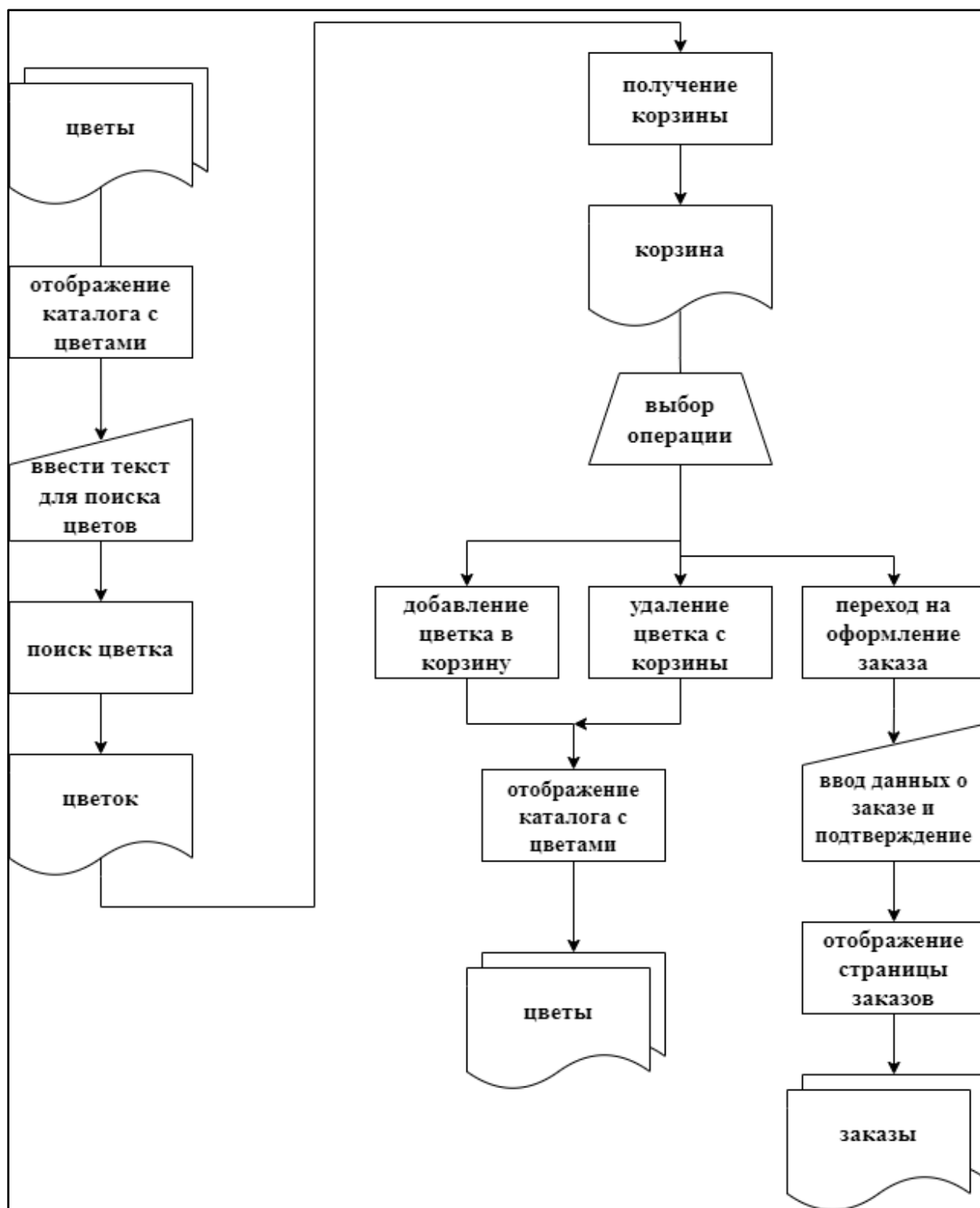


Рисунок 2.2 – Схема данных программного средства

2.3 Инфологическая модель базы данных

Основной целью инфологического проектирования является построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создается без ориентации

на какую-либо конкретную СУБД и модель данных. Конкретный вид и содержание инфологической модели базы данных определяется выбранным для этого формальным аппаратом [6].

Исходя из необходимости использования в проектируемом приложении базы данных, разработаем ее инфологическую модель. Для создания данной модели возьмем за основу предметную область проекта. Предметная область разрабатываемого программного средства включает в себя следующие сущности и их атрибуты:

а) Пользователь:

- 1) Идентификатор.
- 2) Имя.
- 3) Фамилия.
- 4) Адрес электронной почты.
- 5) Номер телефона.
- 6) Адрес.
- 7) Роль.

б) Заказ:

- 1) Идентификатор.
- 2) Идентификатор корзины товаров.
- 3) Идентификатор пользователя.
- 4) Имя покупателя.
- 5) Номер телефона.
- 6) Адрес доставки.
- 7) Тип оплаты.
- 8) Комментарий.

в) Выбранный цветок:

- 1) Идентификатор.
- 2) Идентификатор цветка.
- 3) Количество.

г) Корзина товаров:

- 1) Идентификатор.
- 2) Идентификатор выбранных цветков.
- 3) Идентификатор пользователя.

д) Цветок:

- 1) Идентификатор.
- 2) Название.
- 3) Цена.
- 4) Фото.
- 5) Описание.
- 6) Категория.

Инфологическая модель базы данных включает в себя:

– Описание информационных объектов или понятий предметной области и связей между ними.

– Ограничения целостности, то есть требования к допустимым значениям данных и к связям между ними.

2.4 Спецификация функциональных требований

Для детализации функций рассмотрим основные требования, предъявляемые к каждой функции программного средства как с точки зрения внутренней организации системы, так и с точки зрения взаимодействия системы с пользователем.

При разработке спецификаций функциональных требований кроме всего прочего учитывались особенности платформы разработки.

2.4.1 Функция регистрации

Функция регистрации должна быть реализована с учетом следующих требований:

- Процесс регистрации инициируется пользователем системы (на рисунке 2.1 представлен в виде роли «Гость»).
- Ввод своего имени для системы, адреса электронной почты, а также пароль с его повторением является обязательным.
- Правильность и стойкость введенного пароля должна быть проверена при помощи встроенных инструментов разработки.
- Вывод сообщения при вводе некорректных данных с просьбой повторить попытку.
- Хранение пароля допускается только в хешированном виде.
- Функция смены пароля должна быть предусмотрена и после регистрации.

2.4.2 Функция аутентификации

При реализации функции аутентификации следует учесть следующие требования:

- Инициатором является пользователь, при этом ему необходимо предоставить имя пользователя и пароль, заданные при регистрации.
- Функция повторной аутентификации пользователя без необходимости ввода какой-либо информации должна быть реализована.

2.4.3 Функция просмотра каталога

Данная функция должна быть реализована с учетом следующих требований:

- Наличие разделения по страницам каталога цветов, показывается определенное количество цветов на одной странице.
- Цветок в каталоге должен содержать имя, цену, краткое описание, фотографию и кнопку для добавления его в корзину.

- Фильтры по диапазону цен и по категориям цветков должны быть реализованы.

- Сортировка по возрастанию и убыванию цены, в алфавитном и против алфавитного порядка по названиям должна быть реализована.

- Нажатие левой кнопки мыши вызывает переход на страницу с подробной информацией о цветке.

2.4.4 Функция просмотра и редактирования корзины

Данная функция должна быть реализована с учетом следующих требований:

- Функции удаления и изменения количества цветов в корзине должны быть предоставлены пользователю.

- Корзина должна сохранять свое состояние при перезагрузке страницы.

2.4.5 Функция оформления заказа

Данная функция должна быть реализована с учетом следующих требований:

- Заполнение информации о заказе для авторизованного пользователя с предоставлением функции редактирования этой информации.

- Ввод адреса доставки, адреса электронной почты, выбор способа оплаты и номер телефона, как обязательных полей, и комментария к заказу, по выбору пользователя.

- Ввод адреса должен быть предусмотрен как вручную, так и при помощи выбора точки на карте.

- Выбор способа оплаты должен состоять из оплаты онлайн при помощи карты, оплаты курьеру картой и оплаты курьеру наличными.

- Ввод в поле номера телефона должен позволять вводить только цифры и соответствовать формату белорусского номера.

- Подтверждения заказа сопровождается отправлением письма на указанную электронную почту.

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

3.1 Разработка архитектуры программного средства

На основании анализа вариантов проектирования приложения для различных платформ было принято решение выбрать основной для разработки платформу веб-приложений.

Классической архитектурой для приложений данного типа является двухзвенная клиент-серверная архитектура. Схема такой архитектуры представлена на рисунке 3.1. В данном случае единственная задача сервера – возврат по запросу статической страницы с некоторой информацией. Однако стоит учитывать, что абсолютное большинство ресурсов сети Интернет реализованы с высокой степенью интерактивности, то есть предоставляют некоторые элементы, с которыми пользователь может взаимодействовать. Таким образом, клиентская часть приложения, помимо простого отображения информации, должна быть управляемой с помощью пользовательских воздействий. Выбранная программная платформа позволяет это реализовать.

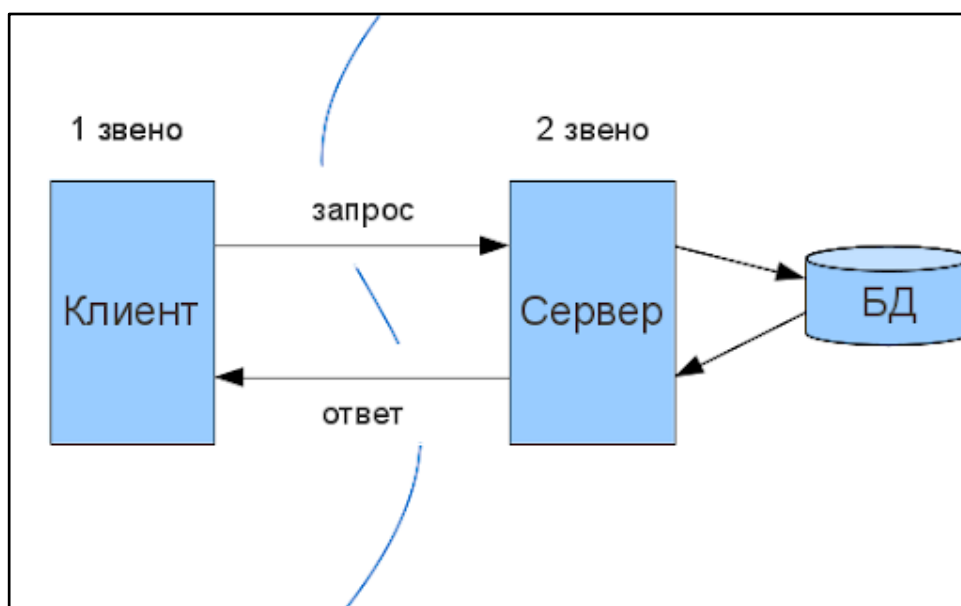


Рисунок 3.1 – Схема двухзвенной архитектуры

Вследствие интерактивности появляется необходимость в хранилище данных, которая обычно решается с помощью реляционных баз данных. В приложении, описываемом данным дипломным проектом, будет использоваться СУБД MS SQL, которая может запускаться как на самом сервере, так и на физически отдельной ЭВМ, что позволит в будущем расширить архитектуру приложения до трехзвенной.

Клиент-серверный шаблон – если есть ограниченное число ресурсов, к которым требуется ограниченный правами доступ большого числа

потребителей, то удобно реализовать клиент-серверную архитектуру. Такой подход повышает масштабируемость и доступность системы. Но при этом сервер может стать узким местом системы, при его недоступности становится недоступна вся система.

Архитектурный шаблон «Клиент-Сервер» имеет ряд достоинств, основными из которых являются:

- Модифицируемость.
- Масштабируемость.

Для представления архитектуры используется диаграмма развертывания.

Диаграмма развертывания отображает физическую структуру будущей системы. Данная диаграмма показывает какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом [7].

Существует два типа узлов:

- Устройства.
- Среды выполнения.

Узел устройства – физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения.

Узел среды выполнения – программный вычислительный ресурс, который работает внутри внешнего узла и который представляет собой сервис, выполняющий другие исполняемые программные элементы.

Для снижения нагрузки предпочтительно использование модели распределенного приложения. Такие приложения разделяются на несколько частей, каждая из которых может выполняться на разных узлах. Данные части взаимодействуют между собой в строго согласованном формате передачи данных.

Клиент-сервер – это сетевая архитектура, в которой сетевая нагрузка распределена между серверами и клиентами. Фактически клиент и сервер – это программное обеспечение. Поскольку одна программа-сервер может выполнять запросы от множества программ-клиентов, ее размещают на специально выделенной вычислительной машине, настроенной особым образом, как правило, совместно с другими программами-серверами, поэтому производительность этой машины должна быть высокой. Из-за особой роли такой машины в сети, специфики ее оборудования и программного обеспечения, ее также называют сервером, а машины, выполняющие клиентские программы соответственно клиентами [8].

Для визуализации элементов и компонентов системы, существующих лишь на этапе ее исполнения, на рисунке 3.2 приводится диаграмма развертывания с изображением элементов, из которых должна состоять система.

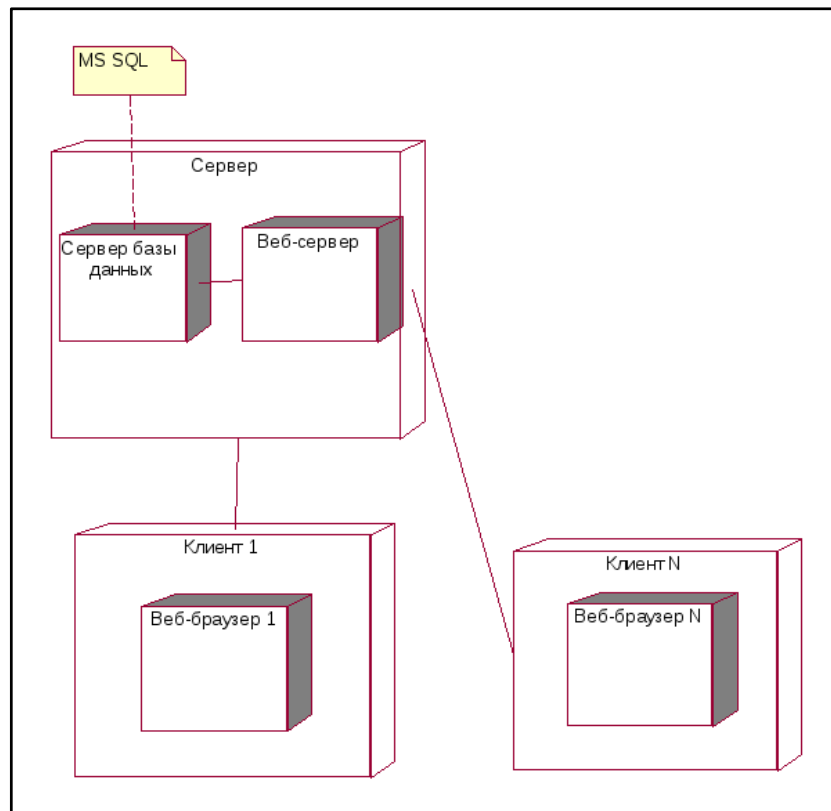


Рисунок 3.2 – Диаграмма развертывания проекта

Для представления архитектуры также используется диаграмма компонентов.

Диаграмма компонентов – статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и тому подобное. На рисунке 3.3 приведена диаграмма компонентов клиентской части проекта, где продемонстрировано непосредственное взаимодействие браузера с проектом разрабатываемого приложения.

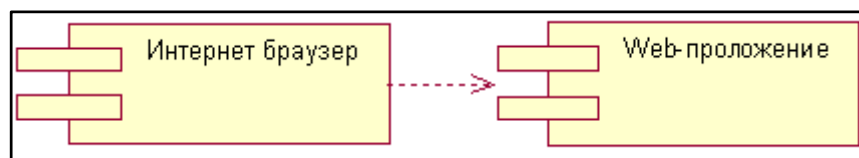


Рисунок 3.3 – Диаграмма компонентов клиентской части

Диаграмма компонентов серверной части проекта приведена на рисунке 3.4, где видно взаимодействие разрабатываемого проекта с сервером и с СУБД, с которым и на основе которых работает проект.

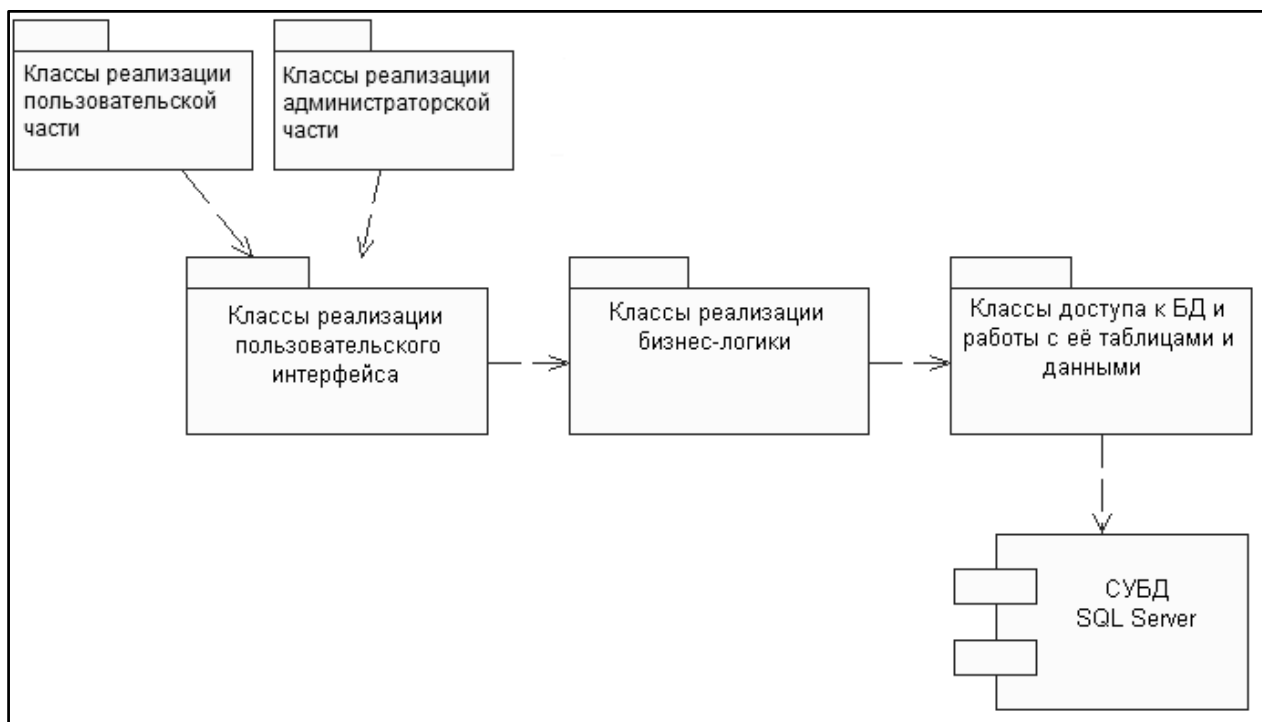


Рисунок 3.4 – Диаграмма компонентов серверной части

3.2 Проектирование модели базы данных

Представление данных в разрабатываемом программном средстве будет осуществляться на уровне базы данных. Для представления данных необходимо спроектировать логическую модель базы данных.

База данных – структурированный организованный набор данных, описывающих характеристики какой-либо физической или виртуальной системы [6].

Логическое проектирование – это процесс создания схемы базы данных на основе выбранной модели. Цель данного этапа проектирования базы данных – преобразование концептуальной модели на основе выбранной модели данных в логическую модель, которая не зависит от особенностей используемой системы управления базами данных.

Для достижения данной цели изначально необходимо выбрать модель данных. Чаще всего выбирается реляционная модель данных в связи с наглядностью табличного представления данных и удобства работы с ними.

Затем определяется набор таблиц исходя из модели. Для каждой сущности модели создается таблица. Таким образом, что каждое имя сущности – имя таблицы. Осуществляется формирование структуры таблиц. Устанавливаются связи между таблицами посредством механизма первичных и внешних ключей.

Следующим этапом является нормализация таблиц. На данном шаге проверяется корректность структуры таблиц, созданных на предыдущем шаге, посредством применения к ним процедуры нормализации. Производится приведение каждой таблицы, по крайней мере, к третьей нормальной форме. В результате нормализации получается очень гибкий проект базы данных, позволяющий легко вносить в нее расширения.

Далее необходимо выполнить проверку логической модели данных на предмет возможности выполнения всех транзакций, предусмотренных программой или пользователями.

Транзакция – набор действий, выполняемых отдельным пользователем или прикладной программой с целью изменения содержимого базы данных.

Тем не менее, разрабатывать модель схемы можно с помощью любых средств, однако затем разработанную модель нужно вручную перевести в формат, используемый СУБД. На даталогическом уровне модель предметной области представляется в привязке к конкретной СУБД и описывает способ организации данных безотносительно их физического размещения.

Модель даталогического уровня разработаем на основании инфологической модели, описание которой приведено в пункте 2.3, представлена на рисунке 3.5. Она состоит из описания всех таблиц, которые будут использоваться в разрабатываемом ПС, а также связей между ними в схематическом виде. Каждая таблица состоит из четырех колонок: первичный ключ, название поля, тип данных и допустимость null значений для данного поля.

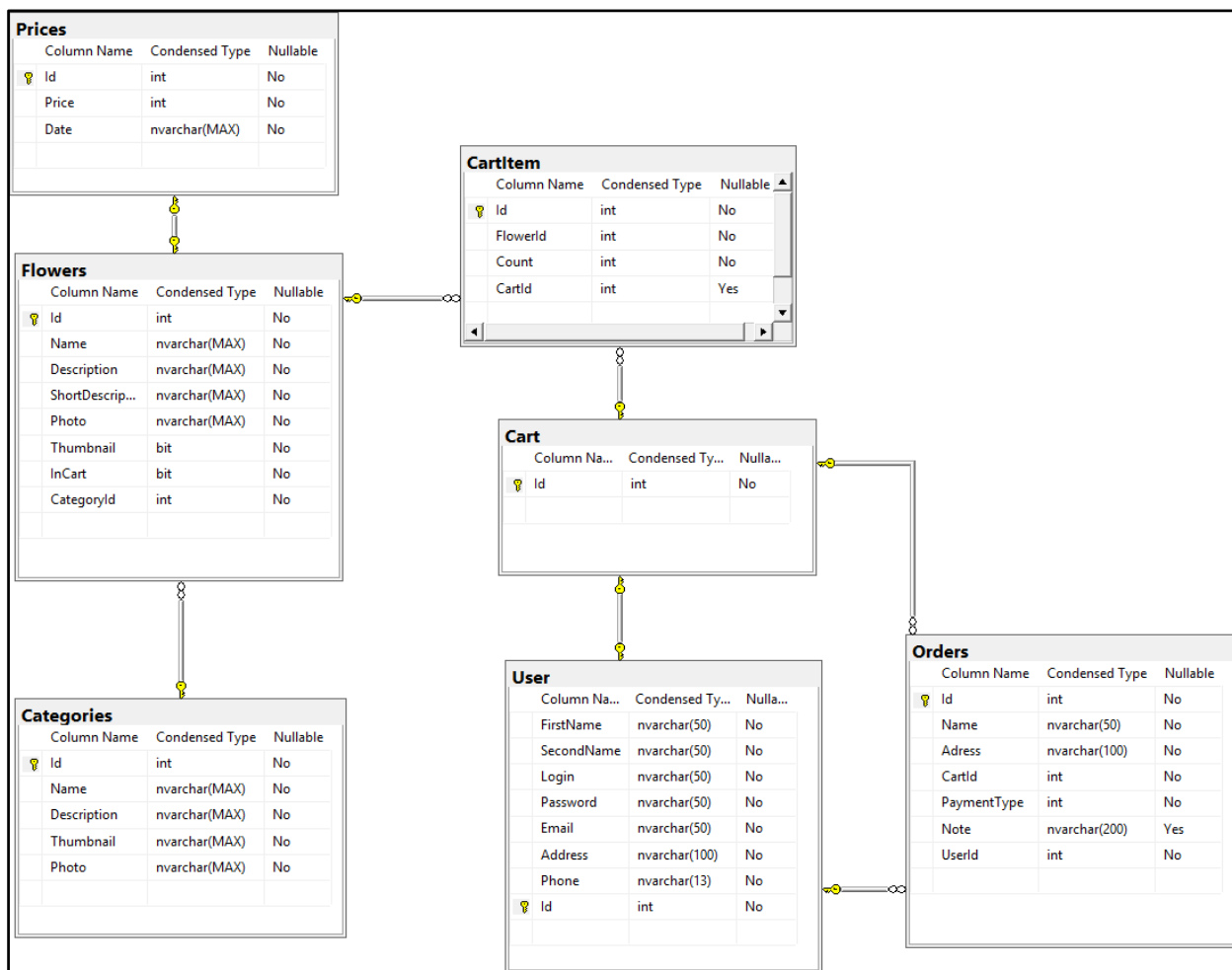


Рисунок 3.5 – Даталогическая модель базы данных

MS SQL Server – это реляционная система управления базами данных с открытым исходным кодом. В настоящее время эта СУБД одна из наиболее популярных в веб-приложениях, наиболее удобно использовать эту базу данных в связи с фреймворком ASP .NET Core, также поддерживаемым и разрабатываемым Microsoft.

MS SQL Server для начался как проект по портированию Sybase SQL Server в 1989 году компанией Sybase, Ashton-Tate и Microsoft. SQL Server выпущен в 1993 году, ознаменовав собой вступление в Windows NT. Написана СУБД на C++ с поддержкой исключений, протестирована на множестве различных компиляторов и работает на различных платформах.

3.3 Проектирование и разработка алгоритмов программного средства

Алгоритм – конечная последовательность действий, призванная привести к результату задачи, для решения которой его используют. Правильно и грамотно разработанный алгоритм гораздо облегчает и ускоряет

получение желаемого итога. Поэтому проектирование и разработка алгоритмов является первостепенной задачей программиста.

В этом подразделе рассмотрим проектирование и варианты реализации нескольких алгоритмов, выполняющих основной функционал.

3.3.1 Проектирование и разработка алгоритма оформления заказа

Диаграмма последовательности метода оформления заказа на рисунке

3.6. Последовательность метода оформления заказа:

- Пользователь является действующим лицом.
- Пользователь вводит необходимую информацию о заказе на странице корзины и оформления заказа.
- Контроллер страницы получает запрос на оформление заказа.
- Контроллер страницы инициирует создание сущности заказа.
- Контроллер страницы формирует запрос на добавление заказа.
- Модуль для работы с БД отправляет запрос к БД.
- База данных осуществляет выполнение запроса на добавление новой записи.
- Модуль для работы с БД получает ответ от БД.
- Контроллер страницы получает результат добавления нового заказа.
- Страница со списком заказов формируется и отображается пользователю.

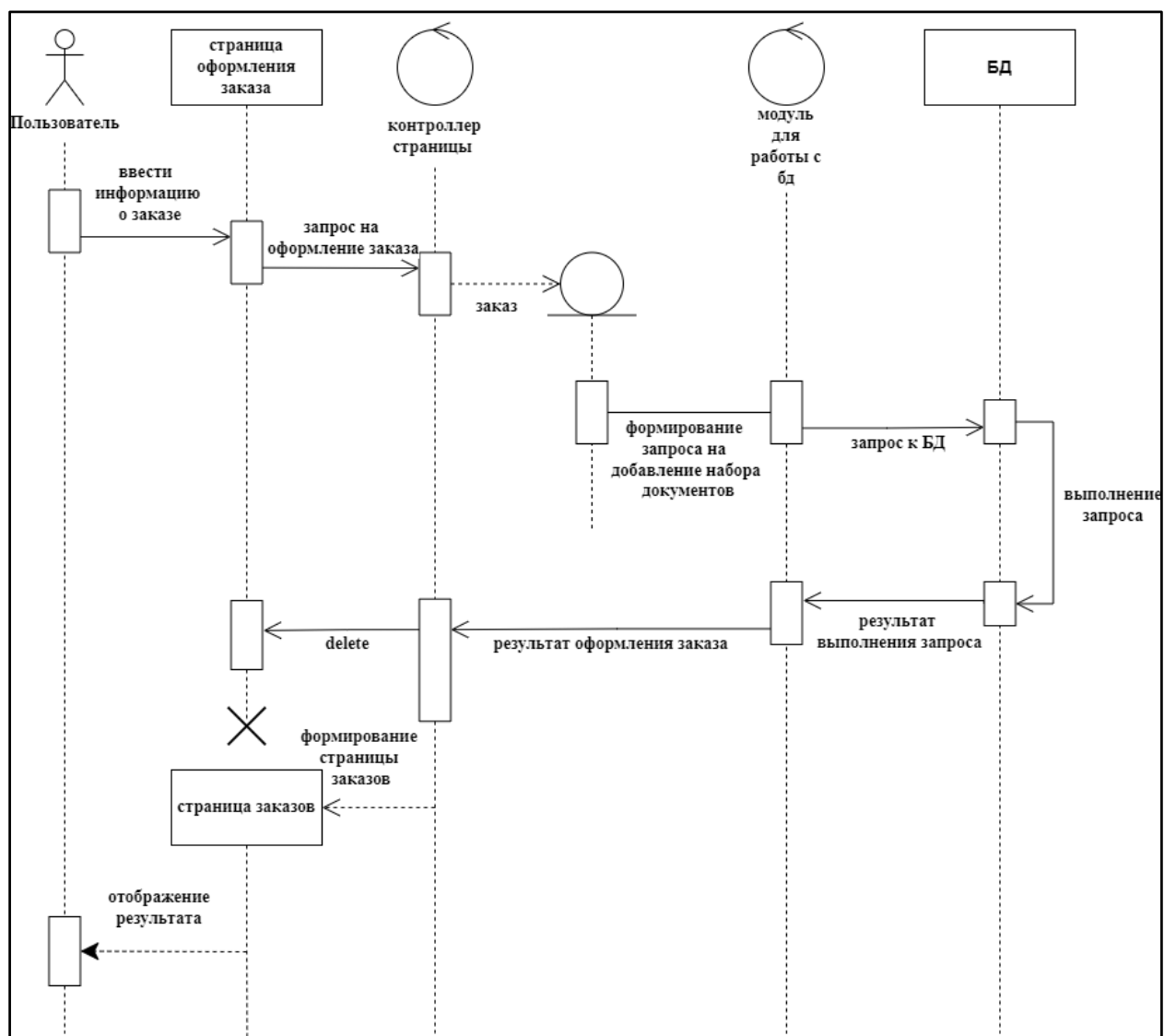


Рисунок 3.6 – Диаграмма последовательности оформления заказа

3.3.2 Проектирование и разработка алгоритма добавления цветка в корзину

Диаграмма последовательности метода добавления цветка в корзину отображена на рисунке 3.7. Последовательность добавления цветка в корзину:

- Пользователь является действующим лицом.
- Пользователь нажимает на кнопку «Добавить в корзину» на соответствующем цветке.
- Контроллер страницы каталога получает запрос на добавление цветка в корзину.
- Контроллер страницы каталога инициирует создание сущности цветка.
- Контроллер страницы формирует запрос на добавление цветка в корзину.
- Модуль для работы с БД отправляет запрос к БД.

- База данных осуществляет выполнение запроса на обновление текущей записи с корзиной.
- Модуль для работы с БД получает ответ от БД.
- Контроллер страницы каталога получает результат добавления нового заказа.
- Страница каталога обновляется и результат выводится на экран.

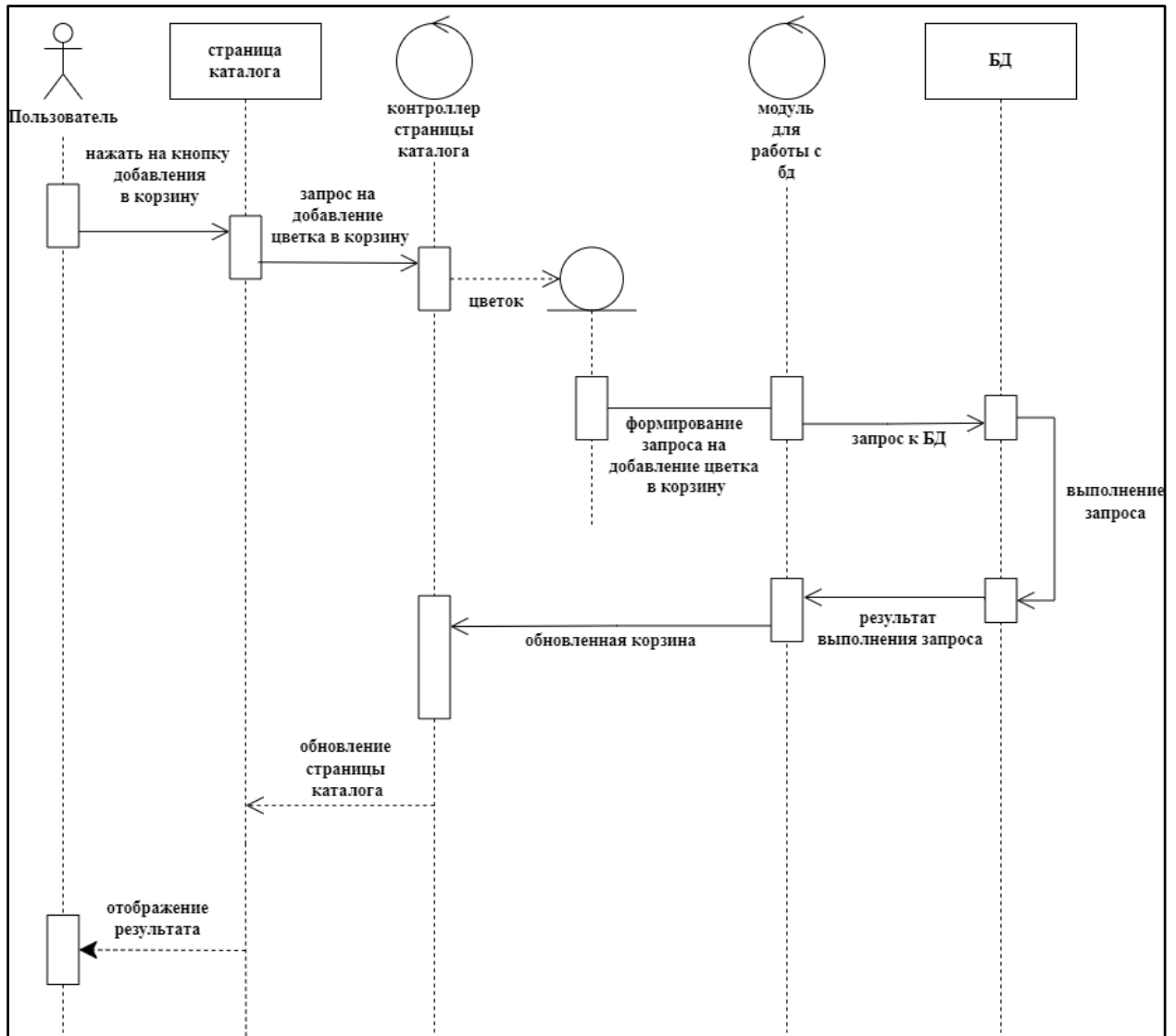


Рисунок 3.7 – Диаграмма последовательности добавления цветка в корзину

4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

Одним из самых важных компонентов в работе программного средства для магазина доставки цветов является работа с заказами. Этой работой занимается класс `OrderController`. Для реализации данного класса был определен перечень возможных действий: получение заказов, добавление заказа, редактирование заказа и удаление заказа. Методы данного контроллера получают данные и передают их на сторону клиента.

Следует рассмотреть подробнее реализацию данного контроллера. К каждому методу контроллера, который возвращает данные клиенту, применяется специальный атрибут `HttpGet`, `HttpPost`, `HttpPut`, `HttpDelete`, указывающий какой именно тип запроса будет обрабатываться методом. Для каждого из контроллеров также определен маршрут с помощью атрибута `Route`.

В зависимости от запроса, пришедшего с клиентской части, вызовется один из методов контроллера. Таким образом, если с клиентской части придет запрос на получение всех заказов, работа программного средства перейдет в метод `GetOrders` класса `OrdersService`. Данный класс предназначен для обработки операций с заказами, извлеченными из базы данных. `getOrders` извлечет все данные о заказах из базы данных, приведет их к виду, нужному клиентскому приложению, учитывая информацию о том, кто делает запрос. Аналогично, `AddOrder` создаст новую запись о заказе, а `UpdateOrder` обновит запись о заказе. `DeleteOrder` удаляет запись о заказе из базы данных.

Одной из немаловажных частей программного средства является работа с базой данных. Для начала необходимо определить в проекте класс модели, с которой производится работа. Эта модель представляет собой те объекты, которые будут храниться в базе данных.

Для взаимодействия с базой данных через `Entity Framework` необходим контекст данных – класс, унаследованный от класса `EntityFrameworkCore`. Свойство `DbSet` представляет собой коллекцию объектов, которая сопоставляется с определенной таблицей в базе данных. При этом по умолчанию название свойства должно соответствовать множественному числу названию модели в соответствии с правилами английского языка. То есть `Order` – название класса модели представляет единственное число, а `Orders` – множественное число. Через параметр `options` в конструктор контекста данных будут передаваться настройки контекста.

Для взаимодействия с базой данных в контроллере определяется переменная контекст данных `ApplicationContext`. Причем поскольку в классе `Startup` в методе `ConfigureServices` контекст данных устанавливается как сервис, то в конструкторе контроллера есть возможность получить переданный контекст данных. Таким образом, в контроллере определены методы, которые добавляют, модифицируют, удаляют и получают все объекты из базы данных.

Таким образом, на примере запроса на получение списка заказов есть возможность продемонстрировать цепочку вызовов в приложении, подытожив иерархию вызовов.

Сначала на клиентской части приложения вызовется метод, который отправит GET-запрос. Данный запрос будет содержать в себе URL на получение заказов. Далее отправленный запрос придет на сервер. Контроллер по полученному URL определит какой именно метод ему нужно вызвать. В вызванном методе контроллера произойдет переадресация вызова в сервис и будет вызван метод «getOrders», который получит из базы данные все доступные текущему пользователю заказы и вернет их клиентской части приложения.

Следует рассмотреть подробнее реализацию клиентской части программного средства. Для отображения списка заказов используется страница «orders.component.ts». В самом начале необходимо определить маршрут данной страницы. Это означает, если к корневому адресу приложения будет добавлено «/orders», произойдет перенаправление на страницу отображения всех заказов.

После этого откроется страница со списком заказов текущего пользователя. В отличие от обычных пользователей, менеджеру здесь будут доступны заказы всех пользователей, а также функции по их редактированию. Со стороны клиентского приложения все функции по работе с заказами реализованы в сервисе «orders.service.ts».

Метод «sendOrder» вызывается на странице оформления заказа при условии, что была нажата кнопка отправления заказа и заполнены все обязательные поля в форме заказа. Метод отправляет POST запрос с информацией о заказе в теле запроса на соответствующий URL, где перехватывается сервером.

Метод «getOrders» вызывается при открытии страницы с заказами. Метод отправляет GET запрос с информацией о текущем пользователе на соответствующий URL, где перехватывается сервером.

В случае, если текущим пользователем является обычный клиент, то он может только просматривать свои заказы, их подробное описание по товарам и статус.

Менеджеру доступны 2 кнопки по редактированию и удалению заказа. Если нажать на кнопку редактирования заказа, откроется страница редактирования заказа. Если нажать на кнопку удаления заказа, откроется модальное окно с подтверждением удаления заказа.

5 ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

Тестирование является одним из важных этапов процесса разработки программного средства. Тестирование – процесс обнаружения ошибок в программном обеспечении.

Стоимость исправления ошибки со временем растет экспоненциально, поэтому важно начинать тестирование на более ранних этапах процесса разработки.

В настоящее время выделяют большое количество видов тестирования. Существует несколько признаков, по которым принято производить классификацию видов тестирования. Обычно выделяют виды тестирования по:

- а) Объекту тестирования:
 - 1) Функциональное.
 - 2) Производительности (нагрузочное тестирование, стресс-тестирование, тестирование стабильности).
 - 3) Конфигурационное.
 - 4) Юзабилити.
 - 5) Безопасности.
 - 6) Локализации.
 - 7) Совместимости.
- б) Знанию внутреннего строения системы:
 - 1) Белого ящика.
 - 2) Серого ящика.
 - 3) Черного ящика.
- в) Степени автоматизации:
 - 1) Ручное.
 - 2) Автоматизированное.
 - 3) Полуавтоматизированное.
- г) По целям и задачам:
 - 1) Регрессионное.
 - 2) Функциональное.
 - 3) Нефункциональное.
- д) Времени проведения тестирования:
 - 1) Альфа.
 - 2) Бета.
- е) Признаку позитивности сценариев:
 - 1) Позитивное.
 - 2) Негативное.
- ж) Степени подготовленности к тестированию:
 - 1) Тестирование по документации.
 - 2) Интуитивное тестирование.

Применение всех видов тестирования в рамках одного проекта целесообразно только в случаях создания очень сложного программного обеспечения [9].

В результате обзора существующих методов и видов тестирования было выбрано ручное тестирование. Был разработан ряд тест-кейсов, проверяющих корректность работы приложения. В таблице 5.1 приведен набор тестовых случаев, использовавшихся при тестировании программного средства.

На протяжении всего времени разработки программного средства осуществлялось регрессионное тестирование. Сутью регрессионного тестирования является воспроизведение ранее выполнявшихся тестов, чтобы удостовериться в том, что добавление новой функциональности или исправление ошибок не привело к появлению новых ошибок [10].

Тестирование проводилось на оборудовании следующей конфигурации:

- Операционная система Windows 10.
- Процессор Intel Core i7-6670QM 2.2 GHz.
- Жесткий диск Apaser AS340 240 GB.
- Графический адаптер NVIDIA GeForce GT 630M.
- ОЗУ 8 GB 1600 MHz DDR3.

Таблица 5.1 – Набор тестовых случаев для тестирования программного средства

№	Название тест-кейса	Шаги	Ожидаемый результат
1	Запуск приложения	1. Запустить веб-браузер. 2. Ввести в адресной строке URL веб-приложения.	1. Отображение в окне веб-браузера главной страницы веб-приложения.
2	Поиск цветка	1. Нажать на кнопку поиска. 2. Ввести название цветка, выставить фильтры и сортировки.	1. Отображение страницы поиска. 2. Отображение страницы поиска с найденными цветками.
3	Фильтрация и сортировка каталога	1. Перейти на страницу каталога. 2. Ввести фильтры, выбрать сортировку. 3. Нажать на кнопку применения фильтров.	1. Отображение страницы каталога. 2. Отображение введенных фильтров и сортировок на странице каталога. 3. Отображение страницы каталога с примененными фильтрами и сортировками

Продолжение таблицы 5.1

№	Название тест-кейса	Шаги	Ожидаемый результат
4	Отмена фильтров в каталоге	<ol style="list-style-type: none"> 1. Перейти на страницу каталога. 2. Ввести фильтры, выбрать сортировку. 3. Нажать на кнопку применения фильтров. 4. Нажать на кнопку отмены фильтров. 	<ol style="list-style-type: none"> 1. Отображение страницы каталога. 2. Отображение введенных фильтров и сортировок на странице каталога. 3. Отображение страницы каталога с примененными фильтрами и сортировками. 4. Отображение страницы каталога с отмененными фильтрами.
5	Добавление цветка в корзину	<ol style="list-style-type: none"> 1. Перейти на страницу каталога. 2. Выбрать цветок и нажать на кнопку добавления в корзину. 3. Перейти на страницу корзины и оформления заказов. 	<ol style="list-style-type: none"> 1. Отображение страницы каталога. 2. Отображение изменения иконки корзины справа сверху с значения «0» на значение «1». 4. Отображение страницы корзины и оформления заказа. 5. Корзина содержит выбранный ранее цветок с количеством равным «1».
6	Оформление заказа	<ol style="list-style-type: none"> 1. Перейти на страницу каталога. 2. Выбрать цветок и нажать на кнопку добавления в корзину. 3. Перейти на страницу корзины и оформления заказов. 4. Ввести всю информацию в форму оформления заказа. 5. Нажать на кнопку подтверждения заказа. 	<ol style="list-style-type: none"> 1. Отображение в окне веб-браузера страницы веб-приложения. 2. Отображение страницы каталога. 3. Отображение страницы корзины и оформления заказа. 4. Отображение страницы со списком заказов. 5. Оформленный заказ находится вверху списка и имеет иконку статуса желтого цвета.

Продолжение таблицы 5.1

7	Отправка сообщения в чате	<ol style="list-style-type: none"> 1. Нажать на кнопку чата в правом нижнем углу экрана. 2. Ввести текст сообщения. 3. Нажать на кнопку отправки сообщения. 	<ol style="list-style-type: none"> 1. Отображение окна чата в правом нижнем углу экрана. 2. Отображение отправленного сообщения с указанием времени отправки
8	Регистрация	<ol style="list-style-type: none"> 1. Нажать на кнопку профиля в правом верхнем углу экрана. 2. Выбрать пункт для перехода на страницу регистрации. 3. Ввести уникальные и ранее не использованные данные в форму регистрации. 4. Нажать на кнопку подтверждения регистрации. 	<ol style="list-style-type: none"> 1. Отображение всплывающего окна под иконкой пользователя. 2. Отображение страницы регистрации. 3. Отображение введенных данных в форму регистрации. 4. Отображение страницы авторизации.
9	Авторизация	<ol style="list-style-type: none"> 1. Нажать на кнопку профиля в правом верхнем углу экрана. 2. Выбрать пункт для перехода на страницу авторизации. 3. Ввести данные зарегистрированного пользователя в форму авторизации. 4. Нажать на кнопку подтверждения авторизации. 	<ol style="list-style-type: none"> 1. Отображение всплывающего окна под иконкой пользователя. 2. Отображение страницы авторизации. 3. Отображение введенных данных в форму регистрации. 4. Отображение главной страницы веб-приложения.

Немаловажным фактором при проведении тестов также является нагрузочное тестирование. Данный вид тестирования дает понимание разработчику о пиковых нагрузках на сайт. На данный момент в нагрузочном тестировании нет большой надобности, так как сервер не имеет сложной вычислительной архитектуры и сможет легко работать при большом количестве одновременных пользователей. Но если количество пользователей будет расти, а функционал приложения будет многократно увеличиваться, то для корректной работы приложения нужно будет провести нагрузочное тестирование.

Все обнаруженные в процессе тестирования ошибки были устранены в процессе отладки и не воспроизводились в следующих версиях приложения.

Таким образом, в последней версии ошибки не обнаружены. В ходе тестирования установлено, что программное средство реализует все функции, описанные в спецификации.

6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Необходимо ввести в адресную строку веб-браузера адрес приложения, после чего пользователь окажется на главной странице веб-приложения. Каждая страница веб приложения состоит из шапки и основного содержимого этой станицы. В шапке находится заголовок сайта, при нажатии на который, произойдет переход на главную страницу сайта. Также в шапке находится навигация по страницам сайта, кнопка для поиска по каталогу, кнопка для перехода на страницу корзины и оформления заказа и кнопка для работы с пользователем. Внешний вид главной страницы отображен на рисунке 6.1.

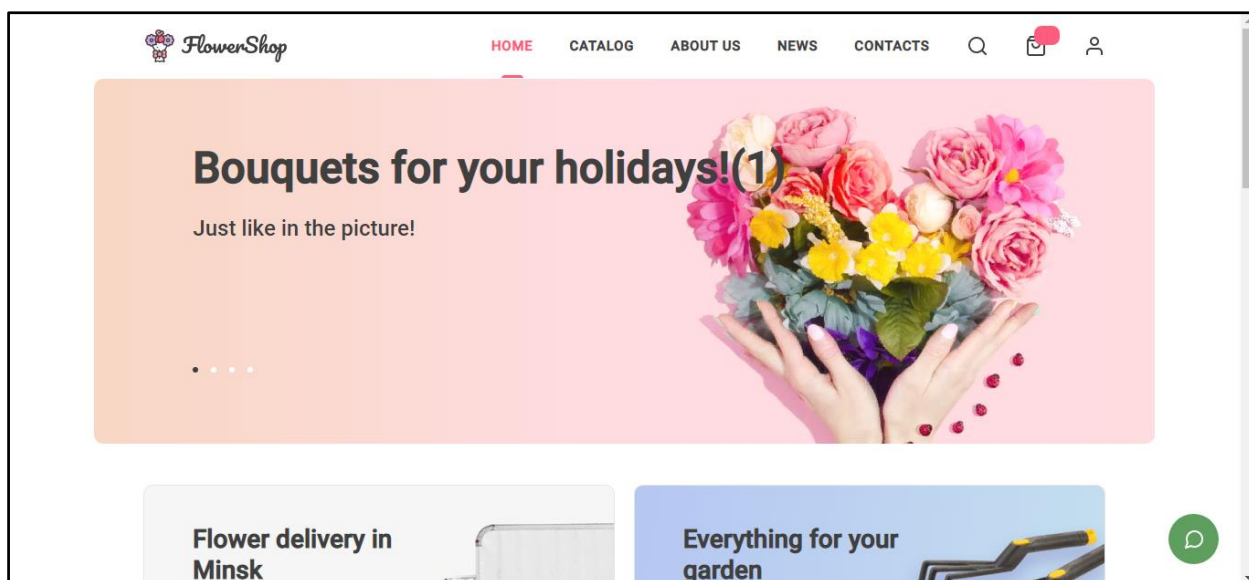


Рисунок 6.1 – Главная страница сайта

Для регистрации в приложении необходимо в меню навигации выбрать пункт для работы с пользователем и далее нажать «REGISTER». Страница регистрации представлена на рисунке 6.2. Для успешной регистрации необходимо ввести все обязательные поля:

- Имя.
- Фамилия.
- Адрес электронной почты.
- Номер телефона.
- Адрес.
- Пароль.
- Повторение пароля.
- Согласие с обработкой персональных данных.

После этого необходимо нажать на кнопку «REGISTER» и произойдет перенаправление на страницу авторизации.

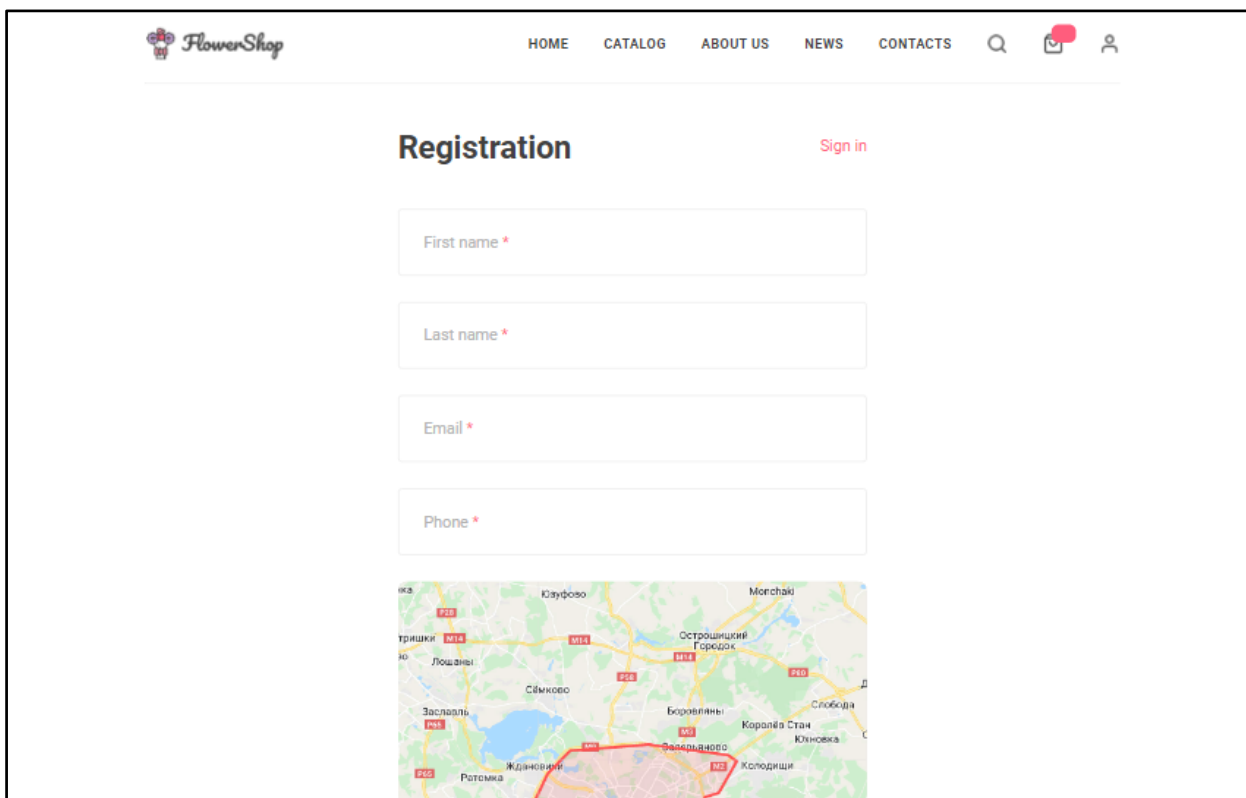


Рисунок 6.2 – Страница регистрации

Для авторизации в приложении необходимо в меню навигации выбрать пункт для работы с пользователем и далее нажать «LOG IN». Внешний вид страницы авторизации приведен на рисунке 6.3. Для успешной авторизации, необходимо заполнить все обязательные поля:

- Адрес электронной почты.
- Пароль.

Вводить необходимо корректные данные, введенные ранее при регистрации. В случае, если введены некорректные данные, будет выведено сообщение об ошибке авторизации.

В случае, если пользователь забыл свои данные, введенные при регистрации, имеется функция восстановления пароля при том, что пользователь помнит введенный адрес электронной почты при регистрации.

Без адреса электронной почты восстановить доступ к аккаунту невозможно. Единственным выходом из данной ситуации будет регистрация нового аккаунта.

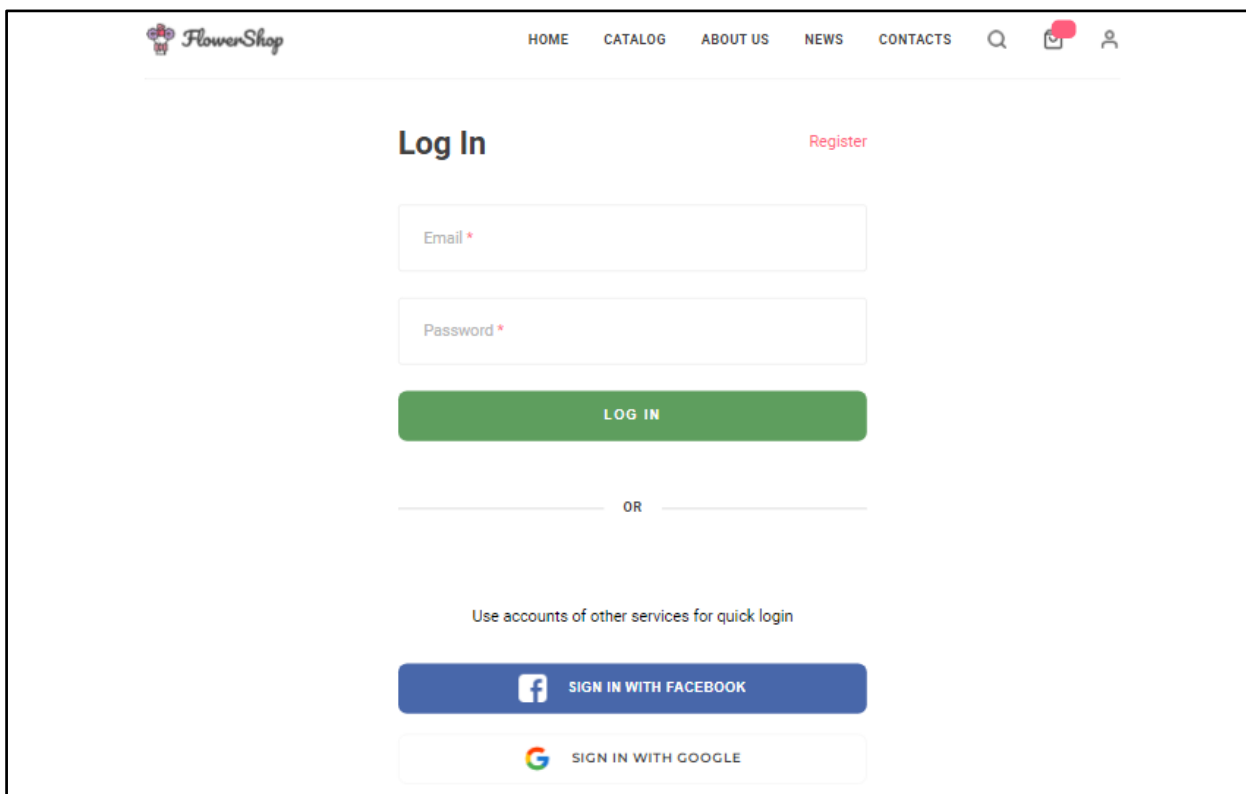


Рисунок 6.3 – Страница авторизации

Для просмотра каталога товаров, необходимо выбрать пункт «CATALOG» в верхнем меню навигации. Внешний вид страницы каталога представлен на рисунке 6.2.

Страница каталога состоит из списка цветов, разделенного по страницам, меню фильтров и выпадающего списка выбора сортировки цветов.

Вверху страницы каталога находится его описание, а также количество найденных цветков.

Каждый цветок в каталоге состоит из фотографии, названия, краткого описания, цены и кнопки для добавления в корзину «ADD TO CART».

Кнопка для добавления в корзину появляется при наведении курсора на цветок в каталоге. Нажатие эту кнопку приведет к добавлению цветка в корзину пользователя или гостя. Сама кнопка при этом изменит свой внешний вид и станет недоступна для повторного нажатия, пока данный цветок содержится в корзине.

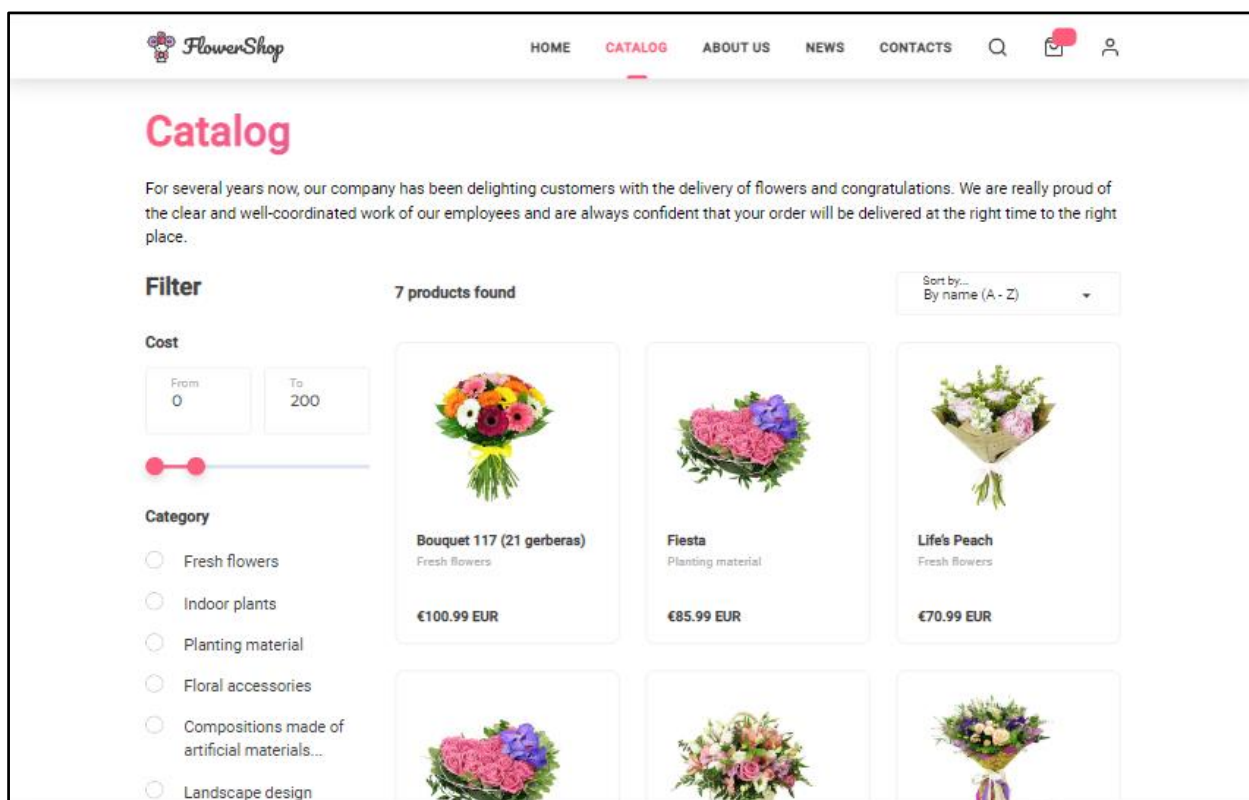


Рисунок 6.4 – Страница каталога цветов веб-приложения

Для того, чтобы активировать фильтр по цене, необходимо либо ввести соответствующие значения цены в поля «From» и «To», либо же изменить положение соответствующих ползунков. После выбора ценового диапазона необходимо подтвердить ввод, нажав на кнопку «APPLY FILTER».

Чтобы активировать фильтр по категории товара, необходимо сделать активными соответствующие пункты списка с категориями, кликнув по нужным маркерам. После выбора нужных категорий необходимо подтвердить ввод, нажав на кнопку «APPLY FILTER».

Для сброса действия всех введенных фильтров необходимо нажать на кнопку «CLEAR FILTER».

В веб-приложении предусмотрено четыре вида сортировок:

- По названию товара в алфавитном порядке, в приложении «By name (A - Z)».
- По названию товара против алфавитного порядка, в приложении «By name (Z - A)».
- По возрастанию цены товара, в приложении «By cost (ascending)».
- По убыванию цены товара, в приложении «By cost (descending)».

Выбор одного из вариантов в соответствующем выпадающем списке приведет к пересортировке списка цветков.

При нажатии левой кнопкой мыши по названию товара произойдет переход на страницу с подробным его описанием. Внешний вид этой страницы отображен на рисунке 6.3.

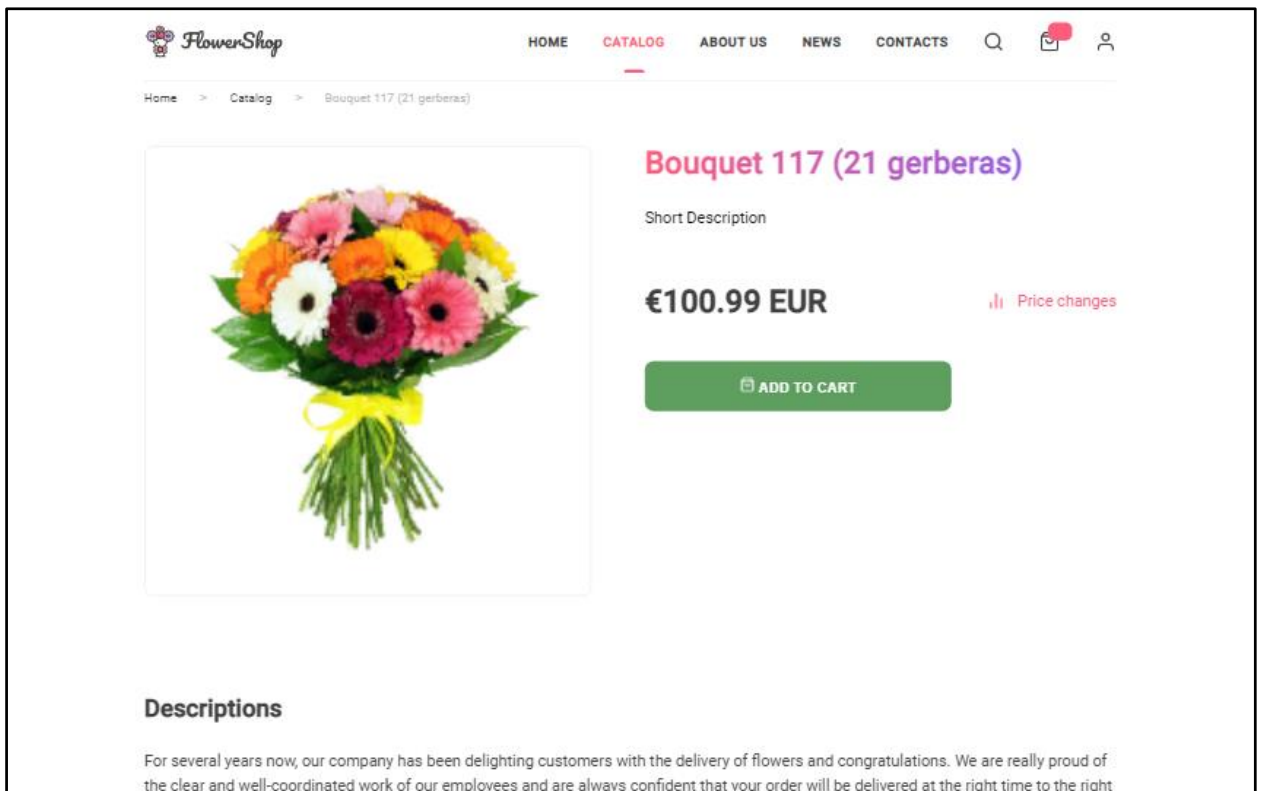


Рисунок 6.5 – Страница подробного описания товара

Данная страница состоит из:

- Фотографии.
- Названия.
- Краткого описания.
- Цены.
- Кнопки перехода на историю изменения цены «Price changes».
- Кнопки добавления в корзину «ADD TO CART».
- Полного описания.

Нажатие на кнопку добавления в корзину приведет к добавлению цветка в корзину пользователя или гостя. Сама кнопка при этом изменит свой внешний вид и станет недоступна для повторного нажатия, пока данный цветок содержится в корзине.

При нажатии на кнопку изменения цены откроется модальное окно, содержащие в себе график, отражающий изменения цены товара во времени. Имеется два варианта диапазона во времени: один месяц и шесть месяцев. Внешний вид модального окна приведен на рисунке 6.4.

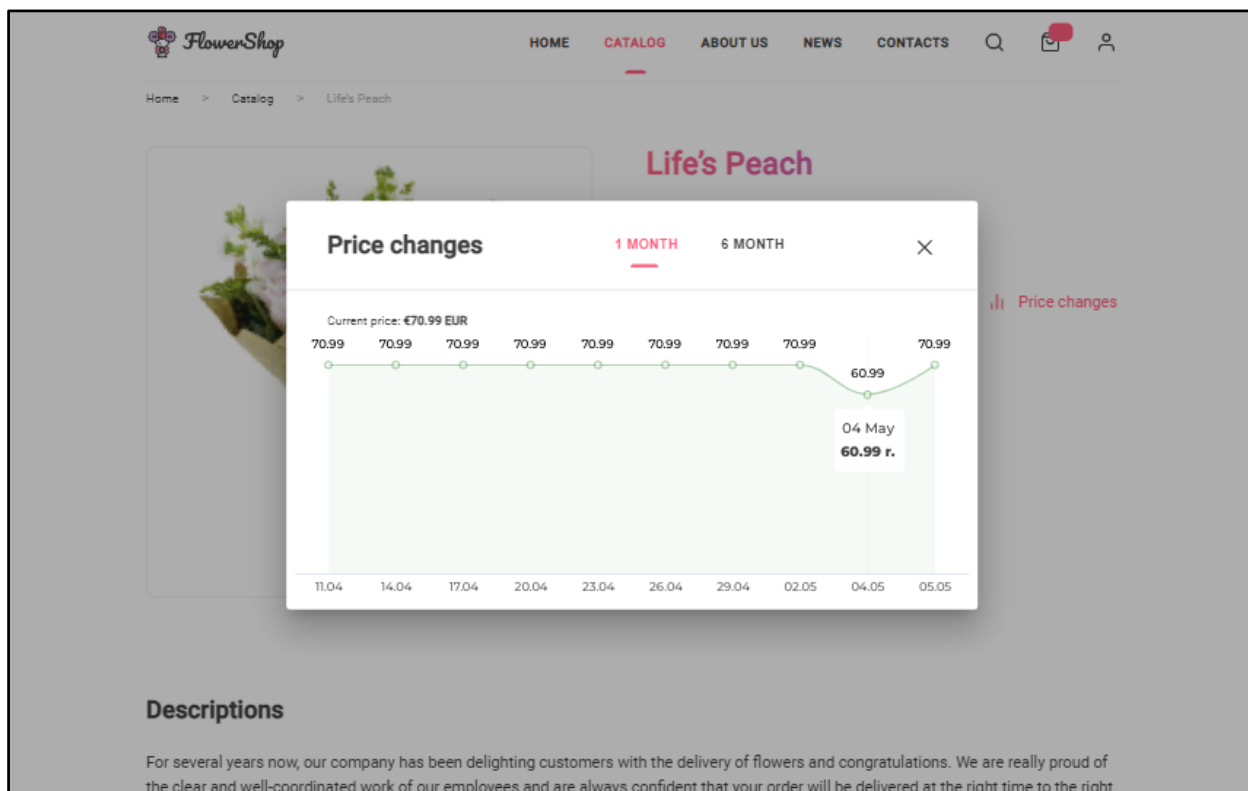


Рисунок 6.6 – Внешний вид модального окна с историей изменения цен

Далее необходимо выбрать нужные цветки и нажать на иконку с корзиной в меню навигации по приложению. Откроется страница корзины и оформления заказа. Содержимое корзины находится в верхней части страницы и представлено на рисунке 6.5.

Чтобы изменить количество определенного товара, необходимо ввести его в соответствующее поле для ввода.

Чтобы удалить определенный товар из корзины, необходимо нажать на кнопку с иконкой крестика в правой части контейнера товара.

Чтобы удалить несколько товаров, необходимо выбрать их посредством нажатия на соответствующее поле в левой части контейнера товара.

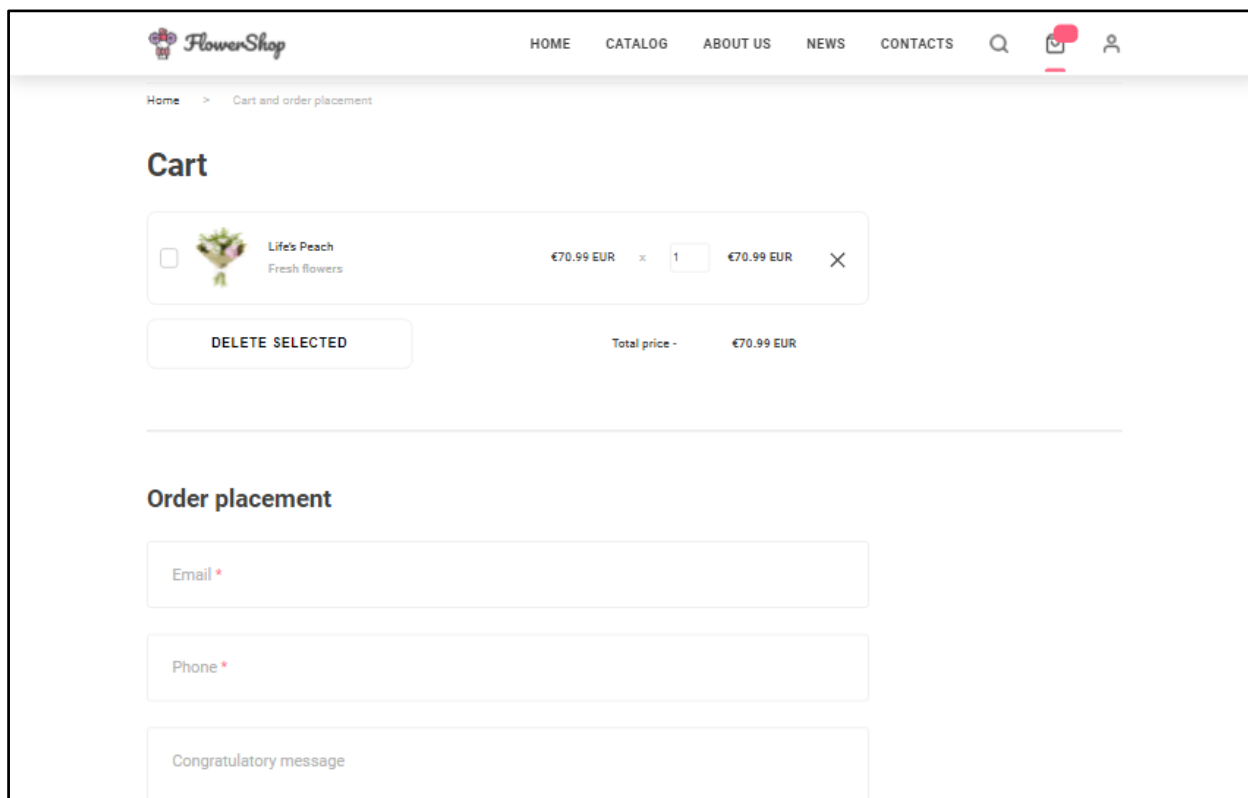


Рисунок 6.7 – Страница корзины и оформления заказа

Для успешного оформления заказа необходимо заполнить все обязательные поля:

- Адрес электронной почты.
- Телефон.
- Адрес доставки.
- Способ оплаты.

У авторизованного пользователя первые три пункта автоматически заполняются и будут доступны для редактирования. Адрес выбирается как по нажатию на карту, так и по вводу вручную с предложением найденных вариантов.

При нажатии на кнопку «CONFIRM ORDER» произойдет перенаправление на страницу онлайн-оплаты, если был выбран данный вариант, либо на страницу с заказами текущего пользователя, если заказ производил авторизованный пользователь и был выбран иной способ оплаты, либо на главную страницу, если заказ производил гость и был выбран иной способ оплаты.

При нажатии на кнопку «CONTINUE SHOPPING» произойдет перенаправление на страницу каталога.

Страница со списком заказов представлена на рисунке 6.6.

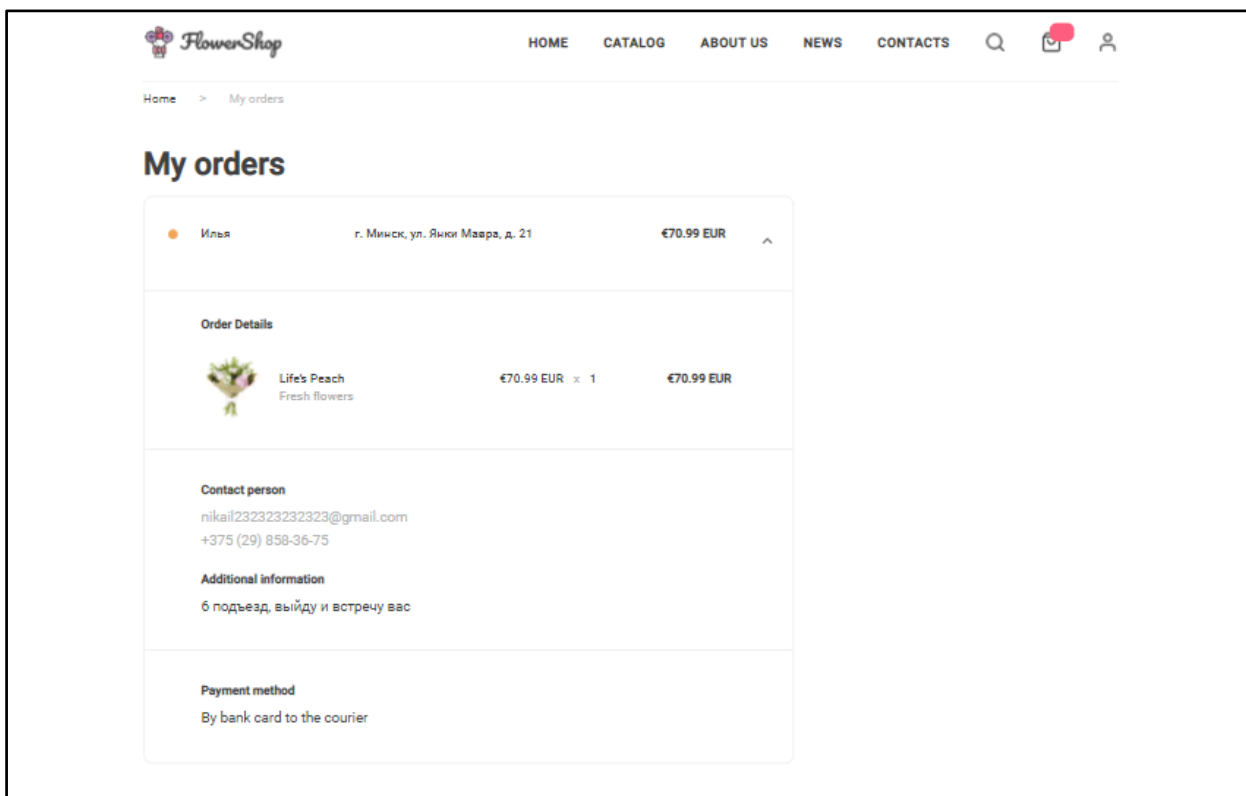


Рисунок 6.8 – Страница со списком заказов

Страница содержит список всех выполненных и выполняемых заказов для текущего пользователя.

Каждый заказ содержит следующую информацию:

- Статус.
- Адрес доставки.
- Суммарная цена заказа.
- Детальной описание заказа по товарам.
- Адрес электронной почты.
- Номер телефона.
- Дополнительная информация.
- Способ оплаты.

После перехода на эту страницу также необходимо подтвердить заказ по электронной почте.

7 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО СРЕДСТВА ДЛЯ МАГАЗИНА ДОСТАВКИ ЦВЕТОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ ANGULAR И ASP .NET CORE

7.1 Характеристика разработанного по индивидуальному заказу программного средства

Разработанное в дипломном проекте программное средство для магазина доставки цветов с использованием технологий Angular, ASP .NET Core предназначено для использования заказчиком с целью автоматизации работы его магазина, упрощения работы с клиентской базой.

Основные функции программного продукта:

- а) Просмотр каталога с реализацией функций:
 - 1) Фильтрации.
 - 2) Сортировки.
 - 3) Поиска.
 - 4) Добавления товара в корзину.
 - 5) Перехода на страницу товара.
- б) Редактирование корзины.
- в) Оформление заказа, включающее:
 - 1) Ввод адреса доставки.
 - 2) Выбор способа оплаты.
 - 3) Ввод прочей контактной информации.
- г) Онлайн-оплата заказа картой.
- д) Переход на страницу:
 - 1) Новостей магазина.
 - 2) Контактной информации магазина.
 - 3) Списка заказов со статусами заказа.

Разработчиком является ООО «Элилинк Консалтинг», занятое в сфере разработки веб-приложений в области информационных технологий.

Заказчиком и единственным покупателем является ООО «Долина цветов», занятое в сфере продаж цветочной продукции.

Применение программного продукта в производственно-хозяйственной деятельности обеспечит заказчику выполнение следующих задач:

- Отображение в приложении и постоянное обновление информации о наличии товара, проведении различных акций и изменении времени работы.
- Привлечение новых клиентов за счет рекламы программного продукта в сети интернет.
- Ускорение деятельности магазина за счет быстрого и упрощенного взаимодействия с клиентами.

Основной потребностью в использовании программного продукта заказчиком является увеличение популярности магазина и приобретение новых клиентов. Подобные приложения, например, «Розы.бел», имеют порядка 500 уникальных посещений в день и более 1500 простых посещений.

Благодаря использованию программного продукта заказчиком ожидается приток новых клиентов, что увеличит конкурентную способность предприятия на рынке и дальнейшую возможность к расширению.

7.2 Расчет затрат на разработку и цена программного средства, созданного по индивидуальному заказу

Для разработки программного средства для магазина доставки цветов необходимы следующие специалисты: backend-программист, frontend-программист, специалист по тестированию программного обеспечения и бизнес-аналитик. Трудозатраты на разработку программного средства у backend-программиста и frontend-программиста будут равны 140 часам, в то время как у специалиста по тестированию программного обеспечения и бизнес-аналитика они будут равны 50 часам.

Расчет основной заработной платы разработчиков осуществляется по формуле:

$$З_o = K_{\text{пр}} \sum_{i=1}^n З_{\text{ч},i} t_i \quad (7.1)$$

где n – количество человек, задействованных в разработке веб-приложения;

$K_{\text{пр}}$ – коэффициент премий;

$З_{\text{ч},i}$ – часовая заработная плата i -го исполнителя (руб.);

t_i – трудоемкость работ, выполняемых i -ым исполнителем.

Примем количество рабочих часов в месяце равным 168 часам. Расчет затрат на основную заработную плату разработчикам приведен в таблице 7.1. Приведенные оклады являются фактическими окладами в СООО «Элилинк Консалтинг».

Таблица 7.1 – Расчет затрат на основную заработную плату команды разработчиков

Наименование должности работника	Вид выполняемой работы	Месячный оклад, р.	Часов ая ставка , р.	Трудовое мкость работ, ч	Сумма, р.
Backend-программист	Разработка сервера и базы данных программного средства	2400	14,3	140	2002
Frontend-программист	Разработка клиентской части программного средства	2200	13,1	140	1834
Специалист по тестированию программного обеспечения	Ручное и автоматизированное тестирование программного средства	1800	10,7	50	535
Бизнес-аналитик	Анализ критериев принятия программного средства заказчиком, формирование сроков разработки	1800	10,7	50	535
Итого					4906
Премия (25%)					1226,5
Всего основная заработная плата					6132,5

Формирование цены программного средства на основе затрат приведено в таблице 7.2.

Таблица 7.2 Формирование цены программного средства на основе затрат

Наименование статьи затрат	Формула/таблица для расчета	Значение, р.
Основная заработная плата разработчиков	Таблица 7.1	6132,5
Дополнительная заработная плата разработчиков	$З_д = \frac{З_о \cdot Н_д}{100\%}, \quad (7.2)$ <p>где $Н_д$ – норматив дополнительной заработной платы (10%)</p> $З_д = \frac{6132,5 \cdot 10}{100\%}$	613,25
Отчисления на социальные нужды	$P_{соц} = \frac{(З_о + З_д) \cdot Н_{соц}}{100\%} \quad (7.3)$ <p>где $Н_{соц}$ – норматив отчислений от фонда оплаты труда (34,6)</p> $P_{соц} = \frac{(6132,5 + 613,25) \cdot 34,6}{100}$	2334,03
Прочие расходы	$P_{пр} = \frac{З_о \cdot Н_{пр}}{100} \quad (7.4)$ <p>где $Н_{пр}$ – норматив прочих затрат, равен 3%</p> $P_{пр} = \frac{6132,5 \cdot 3}{100}$	183,98
Общая сумма затрат на разработку ($З_p$)	$З_p = З_о + З_д + P_{соц} + P_{пр} \quad (7.5)$ $З_p = 6132,5 + 613,25 + 2333,04 + 183,98$	9262,77

Продолжение таблицы 4.2

Наименование статьи затрат	Формула/таблица для расчета	Значение, р.
Плановая прибыль, включаемая в цену программного средства	$П_{п.с} = \frac{З_p \cdot P_{п.с}}{100}, \quad (7.6)$ <p>где $P_{п.с}$ – рентабельность затрат на разработку программного средства (25%)</p> $П_{п.с} = \frac{9262,77 \cdot 25}{100}$	2315,69
Отпускная цена программного средства	$Ц_{п.с} = З_p + П_{п.с}, \quad (7.7)$ $Ц_{п.с} = 9262,77 + 2315,69$	11578,46

В результате расчетов, приведенных в таблице 7.2, была получена отпускная цена программного средства, равная 11578,46 рублям. Цена получилась выше, чем предлагается фриланс-разработчиками в интернете по результатам поиска Google, и в то же время на порядок меньше, чем у более серьезных фирм-разработчиков. Согласно интернет-источнику «thebestapp.com», рассчитанная примерная отпускная цена программного средства с схожим функционалом будет около 80000 рублей.

7.3 Расчет результата от разработки и использования программного средства, созданного по индивидуальному заказу

7.3.1 Оценка экономического эффекта у разработчика

Для организации-разработчика экономическим эффектом является прирост чистой прибыли, полученной от разработки и реализации программного средства заказчику.

Прибыль программного средства, реализованного организацией-разработчиком по отпускной цене, сформированной на основе затрат на разработку, рассчитывается по следующей формуле 7.8

$$\Delta П_ч = П_{п.с} \left(1 - \frac{Н_п}{100} \right), \quad (7.8)$$

где $П_{п.с}$ – прибыль, включаемая в цену программного средства;

$Н_п$ – налог на прибыль.

Так как СООО «Элилинк Консалтинг» является резидентом Парка высоких технологий, то оно освобождается от налога на прибыль. Поэтому получим соответствующее значение:

$$\Delta\Pi_{\text{ч}} = 2315,69 \cdot \left(1 - \frac{0}{100}\right) = 2315,69 \text{ р.}$$

7.3.2 Оценка экономического эффекта у пользователя

Организация-заказчик приобрела программный продукт с целью увеличить количество заказов на их продукцию. Планируется покрывать расходы на приобретение программного средства за счет увеличения чистой прибыли от увеличения количества заказов.

Согласно данным заказчика, средний чек покупки в магазине равен 35 рублям, а среднее количество продаж за год равно 1210. За счет внедрения программного продукта, заказчик планирует увеличить количество продаж до 2100 за счет Seo-продвижения веб-приложения в интернете. Выручка для организации-заказчика рассчитывается по формуле 7.9:

$$\Delta B_p = \text{Ч}_c \cdot (N_{\text{п.в.}} - N_{\text{д.в.}}) \quad (7.9)$$

где $N_{\text{д.в.}}$ – количество заказов до внедрения программного средства;

$N_{\text{п.в.}}$ – количество заказов после внедрения программного средства;

Ч_c – средний чек закупки в магазине.

Примем количество заказов до внедрения программного средства равным 1210, количество заказов после внедрения программного средства равным 2100, средний чек закупки в магазине равным 35 рублям. Подставив соответствующие значения в формулу 7.9, получим:

$$\Delta B_p = 35 \cdot (2100 - 1210) = 31150 \text{ р.}$$

Вычисленное значение является годовым приростом выручки для организации-заказчика. Для расчета чистой годовой прибыли используется формула 7.10:

$$\Delta\Pi_{\text{ч}} = \Delta B_{\text{ч}} \cdot \left(1 - \frac{\text{НДС}}{100}\right) \cdot \left(\frac{P_{\text{пр}}}{100}\right) \cdot \left(1 - \frac{H_{\text{п}}}{100}\right) \quad (7.10)$$

где $\Delta B_{\text{ч}}$ – годовой прирост выручки;

НДС – значение налога на добавленную стоимость;

$P_{\text{пр}}$ – рентабельность продаж;

$H_{\text{п}}$ – налог на прибыль.

Годовая выручка рассчитана по формуле 4.9 и равно 7441,5 рублям. Ставка налога на добавленную стоимость составляет 20%. Рентабельность продаж, согласно данным заказчика, равна 30%. Налог на прибыль составляет 18%. Подставляя описанные ранее значения в формулу 4.10, получаем значение прироста чистой прибыли для заказчика программного обеспечения:

$$\Delta\Pi_{\text{ч}} = 31150 \cdot \left(1 - \frac{20}{100}\right) \cdot \left(\frac{30}{100}\right) \cdot \left(1 - \frac{18}{100}\right) = 6130,32 \text{ р.}$$

Посчитанное значение прироста чистой прибыли также будет являться экономическим эффектом для организации-заказчика от внедрения программного средства.

7.4 Расчет показателей экономической эффективности разработки и использования программного средства

Для организации-разработчика программного средства оценка экономической эффективности разработки осуществляется с помощью расчета простой нормы прибыли (рентабельности инвестиций (затрат) на разработку программного средства) по формуле 7.13:

$$P_{\text{и}} = \frac{\Delta\Pi_{\text{ч}}}{Z_{\text{р}}} \cdot 100\% \quad (7.11)$$

Подставив значения, посчитанные в разделах 7.2 и 7.3 по формулам 7.5 и 7.8, в формулу, получим следующий результат:

$$P_{\text{и}} = \frac{2315,69}{9262,77} \cdot 100\% = 25\%$$

Для расчета показателей экономической эффективности разработки и использования приложения необходимо полученные суммы результата и затрат по годам привести к единому моменту времени – расчетному году (2022 г.) путем умножения результатов и затрат за каждый год на коэффициент дисконтирования(α_t) который рассчитывается по формуле:

$$\alpha_t = \frac{1}{(1 + E_{\text{н}})^{t-1}}, \quad (7.12)$$

где $E_{\text{н}}$ – требуемая норма дисконта, 15%;

t – номер года, результаты и затраты которого приводятся к расчетному (2022 – 1, 2023 – 2, 2024 – 3).

$$\alpha_1 = \frac{1}{(1+0,15)^0} = 1$$

$$\alpha_2 = \frac{1}{(1+0,15)^1} = 0,87$$

$$\alpha_3 = \frac{1}{(1+0,15)^2} = 0,76$$

Расчет рентабельности инвестиций без учета фактора времени производится по формуле:

$$P_{и} = \frac{П_{чср}}{\sum_{t=1}^n 3_t} \cdot 100 \%, \quad (7.13)$$

где $П_{чср}$ – среднегодовая чистая прибыль, полученной от использования разработанного программного средства которая определяется по формуле:

$$П_{чср} = \frac{\sum_{i=1}^n П_{чt}}{n}, \quad (7.14)$$

где $П_{чt}$ – чистая прибыль, полученная в году t , руб.

По формуле 4.13 рассчитана среднегодовая чистая прибыль. Подставив значения, получим:

$$П_{чср} = \frac{6130,32 + 5333,38 + 4659,04}{3} = 5375,25 \text{ руб.}$$

Таблица 7.3 – Расчет показателей экономической эффективности разработки и реализации программного средства

Показатель	Значение по годам расчетного периода		
	2022	2023	2024
Результат			
1. Прирост чистой прибыли от реализации, р.	6130.32	6130.32	6130.32
2. Дисконтированный результат, р.	6130.32	5333,38	4659,04
Затраты			
3. Инвестиции в разработку программного средства, р.	11578,46	—	—
4. Дисконтированные инвестиции, р.	11578,46	—	—
5. Чистый дисконтированный доход по годам, р.	-5448,14	5333,38	4659,04

Продолжение таблицы 7.3

Показатель	Значение по годам расчетного периода		
	2022	2023	2024
6. Чистый дисконтированный доход нарастающим итогом, р.	-5448,14	-114,76	4544,28
Коэффициент дисконтирования, доли единицы	1	0,87	0,76

Таким образом, рентабельность инвестиций, высчитанная по формуле 7.12 получается равна:

$$P_{\text{и}} = \frac{5375,25}{11578,46} \cdot 100\% = 46,42\%$$

Полученная рентабельность говорит о том, что организации-заказчику будет выгодно вложить свои средства в внедрение разрабатываемое программное средство.

Срок окупаемости без учета фактора времени определяется по формуле 7.15:

$$T_{\text{ок}} = \frac{\sum_{t=1}^n 3_t}{\Pi_{\text{ср}}}, \quad (7.15)$$

Подставив значения в формулу, получим:

$$T_{\text{ок}} = \frac{11578,46}{5375,25} = 2,15$$

В результате технико-экономического обоснования разработки и использования программного средства были получены результаты для организации-разработчика:

- Общая сумма затрат на разработку составила 9262,77 рубля.
- Цена программного средства составила 11578,46 рубля.
- Экономический эффект от разработки составил 2315,69 рубля.

Также были рассчитаны результаты от внедрения программного средства организацией-заказчиком:

- Чистый дисконтированный доход составил 5375,25 рубля.
- Уровень рентабельности составил 46,42%.

Таким образом, использование программного средства является экономически эффективным и инвестиции в его разработку целесообразно осуществлять.

ЗАКЛЮЧЕНИЕ

Главной целью данного проекта является разработка качественного и надежного ПО для автоматизации работы магазина по доставке цветов.

В ходе работы над дипломным проектом были проанализированы программные средства для аналогичных магазинов по доставке цветов, исследованы различные подходы к решению задач, связанных с разработкой веб-приложений.

В процессе разработки был произведен анализ предметной области. Также были изучены литературные источники, исследованы существующие аналоги. В результате были выделены существующие недостатки и преимущества аналогов и впоследствии учтены при разработке функциональных требований.

Проектирование программного средства производилось на основе разработанных функциональных требований. В программном средстве были успешно реализованы функции:

а) Просмотр:

- 1) Каталога.
- 2) Новостей магазина.
- 3) Контактной информации магазина.
- 4) Списка заказов со статусами заказа.

б) Редактирование корзины.

в) Оформление заказа.

г) Онлайн-оплата заказа картой.

Согласно функциональным требованиям были разработаны тестовые сценарии, которые были успешно пройдены в ходе тестирования программного средства.

На завершающем этапе было составлено подробное руководство использования программного средства, которое позволяет быстро освоить работу с приложением.

Также в ходе работы над дипломным проектом была рассмотрена экономическая сторона разработки и рассчитана экономическая целесообразность данного проекта. Полученные результаты свидетельствуют о том, что разработка программного средства является рентабельной.

Нужно отметить, что данное программное средство имеет большой потенциал для дальнейшего развития и совершенствования путем добавления нового функционала. Например, добавить функцию сохранять понравившийся товар в раздел «FAVOURITES», чтобы в дальнейшем всегда иметь к нему доступ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Обзор принципов построения интернет-магазинов [Электронный ресурс]. – 2022 – Режим доступа: https://studbooks.net/614219/informatika/obzor_printsipov_postroeniya_internet_magazinov – Дата доступа: 17.05.2022.
- [2] Полезные функции браузера Google Chrome [Электронный ресурс]. – 2022 – Режим доступа: <https://www.google.by/intl/ru/chrome/browser-features/> – Дата доступа: 17.05.2022.
- [3] Общие сведения об ASP .NET Core [Электронный ресурс]. – 2022 – Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-5.0> – Дата доступа: 17.05.2022.
- [4] Преимущества Angular [Электронный ресурс]. – 2022 – Режим доступа: https://club.cnews.ru/blogs/entry/samye_populyarnye_sajty_i_vebprilozheniya_na_angular_kotorymi_polzuetsya_ves_mir – Дата доступа: 17.05.2022.
- [5] Ambler, S. W. UML 2 Deployment Diagrams: An Agile Introduction [Электронный ресурс] / S. W. Ambler. – Agile Modeling, 2014. – Режим доступа: <http://www.agilemodeling.com/artifacts/deploymentDiagram.htm> – Дата доступа: 17.05.2022.
- [6] Куликов, С. С. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. – ЕРАМ Systems, RD Dep., 2017. – http://svyatoslav.biz/software_testing_book/. – Дата доступа: 17.05.2022.
- [7] Коржов, В. Многоуровневые системы клиент-сервер / В. Коржов // Журнал «Сети / Network World». – 1997. – 17 июня. – С. 5.
- [8] Серебряная, Л. В. Объектно-ориентированные технологии программирования и стандарты проектирования : учеб.-метод. пособие / Л. В. Серебряная. – Минск : БГУИР, 2018. – 64 с. : ил.
- [9] Куликов, С. С. Шаблоны корпоративных приложений / С. С. Куликов. – Минск : Четыре четверти, 2015. – 294 с.
- [10] Тестирование программного обеспечения : учеб. пособие / С. С. Куликов [и др.]. – Минск : БГУИР, 2019. – 276 с.: ил.
- [11] О расчетной норме рабочего времени на 2019 год | Новости | Министерство труда и социальной защиты Республики Беларусь. Официальный сайт [Электронный ресурс]. – 2018. – Режим доступа: http://www.mintrud.gov.by/ru/news_ru/view/o-raschetnoj-norme-rabochego-vremeni-na-2019-god_3063/ – Дата доступа: 04.04.2021
- [12] Бахтизин, В.В. Технология разработки программного обеспечения / В.В. Бахтизин, Л.А. Глухова – Минск: БГУИР, 2010. – 267 с.
- [13] Экономическое обоснование проекта по разработке программного обеспечения: Методическое пособие для студентов всех специальностей БГУИР. / Горовой В.Г. и др. – Минск: БГУИР, 2018 г. – 12 с.
- [14] Методические указания по дипломному проектированию для студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий» (ПОИТ). / Бранцевич П.Ю., Бахтизин В.В. –

Минск: БГУИР, 2018. – 82 с.

[15] ASP .NET Core [Электронный ресурс]. – Электронные данные. – Режим доступа : <https://metanit.com/sharp/aspnet5/1.1.php/> – Дата доступа: 17.05.2022.

[16] Флэнаган, Дэвид. Javascript. Подробное руководство. – 6-е изд. – Издательство «Символ плюс», 2013. – 1080 с.

[17] Angular Docs [Электронный ресурс]. – Электронные данные. – Режим доступа : <https://angular.io/docs> – Дата доступа: 17.05.2022.

[18] Горюшкин, А. А. Технико-экономическое обоснование дипломных проектов(работ). Метод. указания для студ. техн. спец. / А. А. Горюшкин, А. В. Грицай, В. Г. Горовой. – Минск, 2020. – 86 с.

[19] Зарплата в ИТ [Электронный ресурс]. – Режим доступа : <https://salaries.dev.by/> – Дата доступа: 17.05.2022.

[20] Глухова Л. А., Фадеева Е. П., Фадеева Е. Е. Основы алгоритмизации и программирования : Лаб. практикум для студ. спец. I-40 01 01 «Программное обеспечение информационных технологий» дневной формы обучения. В 4 ч. Ч.3. – Мн. : БГУИР, 2007. – 51 с.

[21] Куликов, С. С. Работа с MySQL, MS SQL Server и Oracle в примерах : практ. пособие. / С. С. Куликов. – Минск: БОФФ, 2016. – 556 с.

[22] Фаулер, М. Шаблоны корпоративных приложений / М. Фаулер. – М. : Вильямс, 2016. – 544 с.

Приложение А

(обязательное)
Исходный код программы