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 целости 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.

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 находиться 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 возможность 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.
* Возможность 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 всем устройствами, постоянно обновляется и содержит набор полезных инструментов и функций, таких как:

* Безопасное сохранение паролей.
* Синхронизация данных на различных устройствах.
* Удобная работа с вкладками: возможность 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.
* Умная 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 всё 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.
* Встроенное внедрение зависимостей.
* Интеграция различных клиентских платформ, в том числе и Angular.
* Упрощенный и высокопроизводительный модульный конвейер HTTP-запросов.
* Интеграция 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:

* Быстрое написание кода.
* Более чистый код.
* Возможность повторного использования кода.
* Регулярные обновления.
* Быстрое тестирование отдельных частей приложения.
* Двустороннее связывание данных, мгновенно отражающее изменения. состояния веб-страницы в пользовательском интерфейсе.
* Высокая производительность.
* Простые шаблоны.
* Хорошая индексация в Google [4].

Подведя 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

После того как был определен стек технологий, которые будут использоваться в процессе написания данного приложения, нужно 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»

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

* Просмотр каталога, включающий сортировки, фильтры, поиск, пагинацию.
* Редактируемая корзина.
* Выбор способа доставки и оплаты при оформлении заказа.

Стоит отметить, что вся расположенная на странице информация легко распознается при первом просмотре и не вызывает затруднений с её поиском. Контактная информация находиться на самом видном месте на верху сайта.

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

* Отсутствие онлайн-чата, что ведет к невозможности уточнить информацию без телефонного звонка.
* Устаревший пользовательский интерфейс.
* Невозможно выбрать адрес доставки по карте.

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»



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:

* Просмотр акций.
* Просмотр каталога, включающий сортировки, фильтры, поиск, пагинацию.
* Редактируемая 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:

* Отсутствует возможность регистрации и авторизации: делает невозможным ускорение процесса оформления заказа.
* Нет возможности уточнить наличие товара без связи с сотрудниками магазина.

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.
* Редактируемая корзина.
* Выбор способа оплаты и доставки.
* Связь с сотрудниками как через социальные сети, так и через онлайн-чат.
* Возможность 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 данные 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:

* просмотр каталога:

а) фильтрация каталога;

б) сортировка каталога;

в) поиск по каталогу;

г) добавление товара в корзину;

д) переход на страницу товара;

* редактирование корзины;
* оформление заказа:

а) ввод адреса доставки;

б) выбор способа оплаты;

в) ввод прочей контактной информации;

* онлайн-оплата заказа картой;
* просмотр новостей магазина;
* просмотр контактной информации магазина;
* просмотр 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:

* удобный и понятный интерфейс программы;
* чат для связи с сотрудниками магазина;
* регистрация нового пользователя;
* авторизация пользователя;
* личный кабинет пользователя;
* возможность выбора 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.

Поскольку одна из основных причин отказа приложения – это некорректное взаимодействие пользователя с операционной системой, необходимо обеспечить возможность эксплуатации программного средства конечным пользователем без предоставления ему административных привилегий.

Также для надежного функционирования программного средства обязательно наличие необходимого уровня опыта и квалификации у конечного потребителя.

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

− процессор Intel Core i7;

− жесткий диск объемом 1 Tб;

− 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 Сore с тактовой частотой 2 ГГц и более;

− оперативная память 4 Гб и более;

− возможность 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 актёров и прецедентов.

Данная диаграмма позволяет описать систему на концептуальном уровне. На диаграмме актёром является множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Актёром 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 (актёрами) [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 авторизированному пользователю.

Регистрация, аутентификация и авторизация – функции, которые доступны для роли «Гость» (пользователь, не зарегистрированный в системе). В приложении будет использоваться Keycloak Authentification, 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.

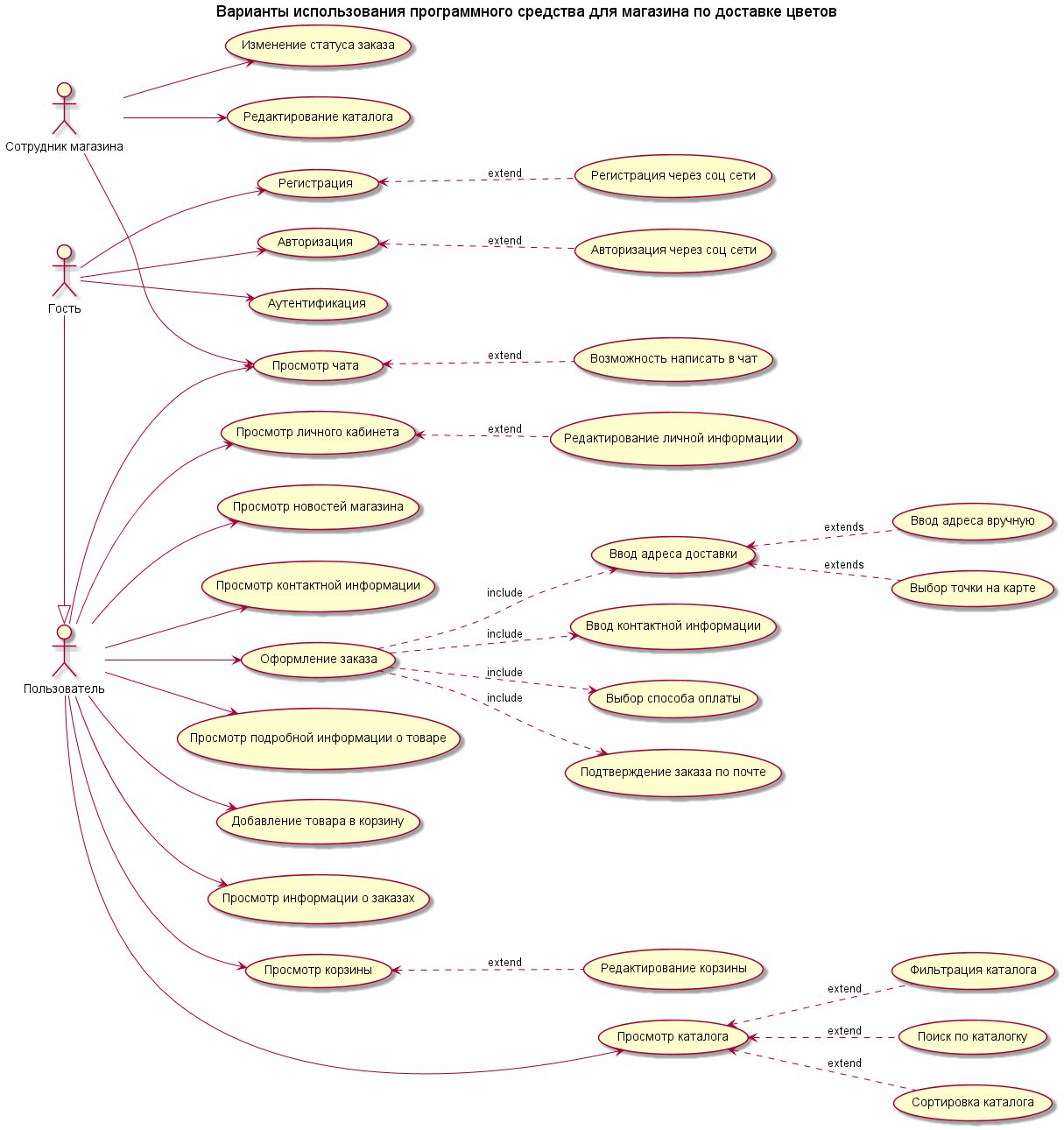


Рисунок 2.1 – Диаграмма вариантов использования программного средства

Рассмотрим детально представленные на рисунке прецеденты.

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

* Просмотр каталога. Сюда входят возможности поиска, сортировки и фильтрации по нему.
* Просмотр личного кабинета. Данная возможность 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 закупаемом товаре и его количестве.
* 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.
* Просмотр информации о заказах. Отвечает за возможность пользователем смотреть список всех его заказов, как выполненных, так и выполняемых в данный момент.
* Просмотр подробной информации о товаре. Необходимо для того, чтобы пользователь мог узнать больше информации о покупаемом товаре, чем имеется в каталоге.
* Просмотр 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 возможность 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 возможность 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 изображён блок процесса, который осуществляет поиск цветов.

4) В результате поиска пользователь получает необходимую ин-формацию 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.

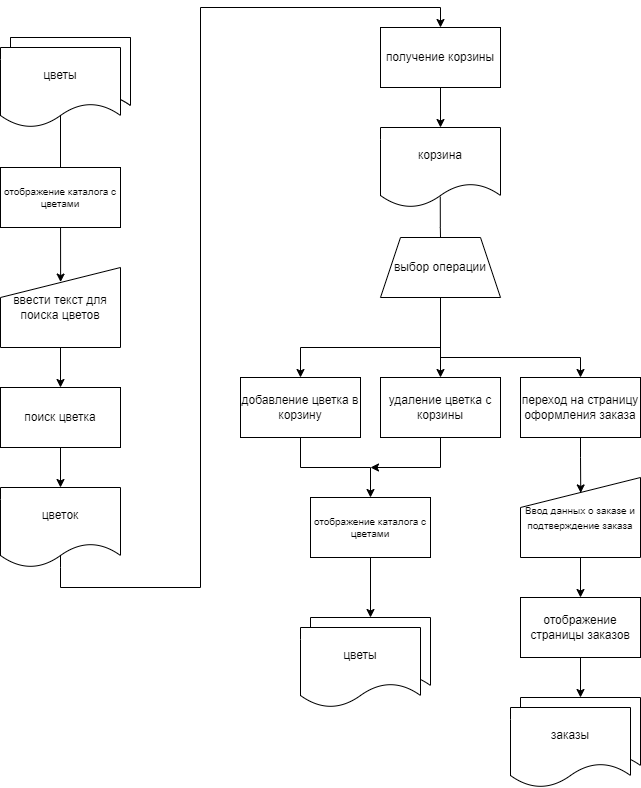


Рисунок 2.2 – Схема данных программного средства

**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 создаётся 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:

* + 1. пользователь:  
       – идентификатор;
       - имя;
       - фамилия;
       - адрес электронной почты;
       - номер телефона;
       - адрес;
       - роль;
    2. заказ:
       - идентификатор;
       - идентификатор корзины товаров;
       - идентификатор пользователя;
       - имя покупателя;
       - номер телефона;
       - адрес доставки;
       - тип оплаты;
       - комментарий;
    3. выбранный цветок:
       - идентификатор;
       - идентификатор цветка;
       - количество;
    4. корзина товаров:
       - идентификатор;
       - идентификаторы выбранных цветков;
       - идентификатор пользователя;
    5. цветок:
       - идентификатор;
       - название;
       - цена;
       - фото;
       - описание;
       - категория;

Инфологическая модель базы данных включает в себя:

– описание информационных объектов или понятий предметной области и связей между ними;

– описание ограничений целостности, то есть требований 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»);
      * для регистрации пользователь обязан предоставить своё имя для системы, адрес электронной почты, а также дважды ввести пароль;
      * правильность и стойкость введенного пароля должна быть проверена при помощи встроенных инструментов разработки;
      * в случае некорректности введенных данных пользователь должен увидеть сообщение об этом с предложением попробовать еще раз;
      * хранение пароля допускается только в хешированном виде;
      * должна быть предусмотрена возможность смены пароля 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;
      * должна быть реализована возможность повторной аутентификации пользователя без необходимости ввода какой-либо информации.

**2.4.3** Функция просмотра каталога

Данная функция должна быть реализована с учетом следующих требований:

* + - * необходимо разделить по страницам все цветы, показывая определенное количество на одной странице;
      * цветок в каталоге должен содержать имя, цену, краткое описание, фотографию и кнопку для добавления его в корзину;
      * должны быть реализованы фильтры по диапазону цен и по категориям цветков;
      * должны быть реализованы сортировки по возрастанию и убыванию цены, в алфавитном и против алфавитного порядка по названиям;
      * при нажатии левой кнопки мыши должен произойти 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:

* + - * необходимо предоставить пользователю возможность удалять или менять количество цветов в корзине;
      * корзина должна сохранять своё состояние при перезагрузке страницы;

**2.4.5** Функция оформления заказа

Данная функция должна быть реализована с учетом следующих требований:

* + - * необходимо автоматически заполнять информацию о заказе для авторизированного пользователя, предоставлять возможность редактировать эту информацию;
      * необходимо запрашивать у пользователя ввод адреса доставки, адреса электронной почты, выбор способа оплаты и номер телефона, как обязательные поля, и комментарий 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;
      * после подтверждения заказа должно высылаться письмо на указанную электронную почту с подтверждением заказа.

# **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 развёртывания.

Диаграмма развёртывания 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 распределённого 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, её 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 её оборудования и программного обеспечения, её также называют сервером, а машины, выполняющие клиентские программы соответственно клиентами [8].

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



Рисунок 3.2 – Диаграмма развёртывания 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 который 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 создаётся 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 неё 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.



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:

* действующим лицом является пользователь;
* 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;
* формируется страница со списком заказов и отображается пользователю.

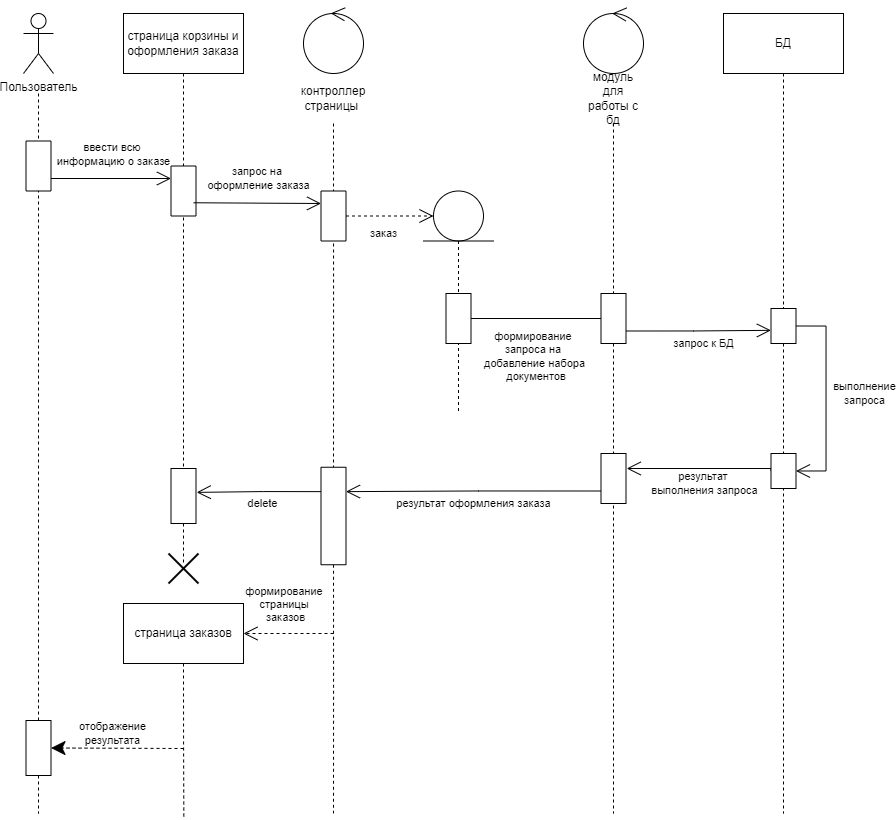


Рисунок 3.6 – Диаграмма последовательности оформления заказа

Список использованных источников

[1] Обзор принципов построения интернет-магазинов [Электронный ресурс]. – 2022 – Режим доступа: https://studbooks.net/614219/informatika/obzor\_printsipov\_postroeniya\_internet\_magazinov

[2] Полезные функции браузера Google Chrome [Электронный ресурс]. – 2022 – Режим доступа: https://www.google.by/intl/ru/chrome/browser-features/

[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

[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

[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  
 [6] Куликов, С. С. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. – EPAM Systems, RD Dep., 2017. – http://svyatoslav.biz/software\_testing\_book/.