

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры полностью)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Корпоративные информационные системы

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка web-представительства для предприятия общественного питания»

Студент

А.И. Гербель

(И.О. Фамилия)

(личная подпись)

Руководитель

канд. пед. наук., доцент Е.В. Панюкова

(ученая степень, звание, И.О. Фамилия)

Консультант

А.В. Москалюк

(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Актуальность темы «Разработка web-представительства для предприятия общественного питания» обусловлена заказом на разработку web-представительства, которое позволит предоставлять информацию о компании и автоматизировать процесс заказа.

Объем выполненной бакалаврской работы – 47 страниц, содержит 24 рисунка, 7 таблиц, 22 литературных источника.

Объектом исследования являются процессы представления информации о продукции и формирования заказа.

Предметом исследования является автоматизация процессов ознакомления клиента с информацией в интернете и формирования заказа.

Во введении описывается актуальность работы, формируется цель и ставятся задачи.

В первой главе дается анализ предприятия и бизнес-процесса, подлежащего автоматизации. Во второй главе проводятся обоснование архитектуры и проектирование информационной системы. В третьей главе описано обоснование выбора средств реализации web-представительства, представлены изображения экранных форм реализованного приложения, а также произведено тестирование.

В заключении приводятся основные выводы, достигнутые в ходе выполнения работы.

Результатом работы является разработанное web-представительство предприятия, способное информировать клиента о компании и помогающее оформить заказ.

Abstract

The title of the graduation work is *Development of a web-representation for a public catering company*.

The object of the study is the processes of presenting information about products and forming an order.

The subject of the study is the automation of the processes of familiarizing the client with information on the Internet and forming an order.

The introduction describes the relevance of the work, forms the goal and sets the tasks. The goal is to develop a web representation for a catering company.

To achieve the goal, the following tasks are identified: analysis of scientific and educational literature; analysis of the business process using existing methodologies and identifying shortcomings; design of conceptual, logical and physical models; analysis of possible technologies for implementing web representation; development of a web application; testing of a web application.

The first chapter provides an analysis of the enterprise and the business process to be automated. In the second chapter, the justification of the architecture and design of the information system are carried out. The third chapter describes the rationale for choosing the means of implementing web-representation, presents images of the screen forms of the implemented application, and also presents testing.

In conclusion, the main conclusions reached in the course of the work are presented.

The result of the work is a developed web-representation of the company, which is able to inform the client about the company and help to place an order.

The graduation work consists of an explanatory note on 46 pages, including 24 figures, 7 tables, the list of 22 references including 10 foreign sources.

Оглавление

Введение.....	3
Глава 1. Функциональное моделирование деятельности предприятия.....	5
1.1 Техничко-экономическая характеристика деятельности предприятия.....	5
1.2 Концептуальное моделирование web-представительства предприятия .	6
Глава 2. Проектирование web-представительства предприятия.....	18
2.1 Обоснование архитектуры проектируемого web-представительства .	18
2.2 Моделирование web-представительства «Donutello»	20
2.3 Проектирование базы данных web-представительства предприятия .	23
2.4 Физическое моделирование web-представительства предприятия	26
Глава 3 Реализация web-представительства предприятия.....	29
3.1 Обоснование выбора технологий реализации web-приложения	29
3.2 Разработка базы данных web-приложения.....	31
3.3 Разработка серверной части web-приложения.....	34
3.4 Разработка клиентской части web-приложения.....	37
3.5 Тестирование	41
Заключение	47
Список используемой литературы и используемых источников.....	48

Введение

На данный момент одним из самых популярных способов продвижения компаний является осуществление информирования потенциальных клиентов о компании в сети интернет. Эту задачу выполняет web-представительство.

«Под веб-представительством компании подразумевается web-страница (или веб-сайт как совокупность веб-страниц), удовлетворяющую следующим критериям:

- веб-страница управляется самой компанией или ее уполномоченным представителем. Это означает, в частности, что содержание страницы (сайта), как и график его изменения, устанавливается компанией, и что другие фирмы не имеют равных прав по использованию этой веб-страницы в своих коммерческих интересах;
- веб-страница используется для коммерческого взаимодействия компании со своими потребителями (распространение информации о товарах и услугах компании, ведение электронной торговли и т. д.);
- веб-страница (веб-сайт) однозначно и без привлечения дополнительной информации идентифицируется другими пользователями интернета как принадлежащая данной компании;
- веб-страница имеет постоянный URL-адрес, а в том случае, если речь идет о веб-сайте, то каждая из его страниц имеет свой URL-адрес, и при этом все страницы связаны гиперссылками» [21].

При этом, даже судя по этим критериям, под определение web-представительства нельзя отнести:

- размещение информации о фирме в телефонном справочнике в сети интернет;
- сайты созданные не по заказу компании и не представляющие ее интересы;
- сообщения на различных форумах о предоставлении коммерческих

услуг.

Актуальность темы определяется заказом от ИП «Юрина Светлана Владимировна».

Объектом исследования являются процессы представления информации о продукции и формирования заказа.

Предметом исследования является автоматизация процессов ознакомления клиента с информацией в интернете и формирования заказа.

Целью данной работы является разработка web-представительства для предприятия общественного питания для того, чтобы предоставлять информацию о компании и взаимодействовать с клиентами онлайн.

Для того чтобы разработать web-представительство, необходимо выполнить следующие задачи:

- проанализировать научную и учебно-методическую литературу, которая необходима для разработки web-представительства;
- провести анализ бизнес-процесса при помощи существующих методологий и выявить недостатки;
- построить концептуальную, логическую и физическую модели;
- проанализировать возможные технологии реализации web-представительства и выбрать подходящие;
- разработать web-приложение используя выбранные технологии реализации;
- провести тестирование web-приложения.

Результатом работы является разработанное web-представительство.

Глава 1. Функциональное моделирование деятельности предприятия

1.1 Техничко-экономическая характеристика деятельности предприятия

Заказчиком для данной работы является Индивидуальное Предприятие, занимающееся изготовлением продукции для заведений общественного питания и доставкой. В качестве клиентов компании выступают юридические лица, общественные заведения, такие как кафе, рестораны, а также столовые. Реализация продукции производится в городе Тольятти.

Организационная структура компании представлена на рисунке 1.



Рисунок 1 – Организационная структура ИП «Юрина Светлана Владимировна»

Услуги, которые предоставляет ИП:

- принятие заказов на продукцию для заведений общественного питания;

- доставка продукции;
- изготовление продукции.

В штате организации состоят:

- бухгалтер;
- шеф-повар;
- повара холодного цеха;
- повара горячего цеха;
- повара-кондитеры;
- курьер;
- уборщик;
- менеджер по работе с заказами.

Работой с заказами на продукцию занимается менеджер по работе с заказами.

На данный момент обработка и формирование заказов происходит вручную, а документация находится на бумажных носителях, что препятствует развитию компании. Также отсутствует информация о компании в сети интернет, что сокращает клиентскую базу. Из этого можно сделать вывод, что организация нуждается в web-представительстве, которое будет предоставлять информацию о компании в интернете, а также позволит автоматизировать процесс оформления заказа.

1.2 Концептуальное моделирование web-представительства предприятия

1.2.1 Обоснование необходимости использования web-представительства

На данный момент оформление заявки клиентом не автоматизировано и происходит лично.

Регистрация заказов производится вручную менеджером, который непосредственно контактирует с клиентом.

Огромный объем данных фиксируется на бумажных носителях, хранится в архивах и не поддается оцифровке, вследствие чего возникают сопутствующие проблемы с анализом информации.

Современные технологии позволяют хранить достаточные объемы данных используя электронные носители, что позволяет снизить трудозатратность и сложность выполнения определенных процессов компании.

Также стоит учитывать, что автоматизация оформления заказа при помощи web-представительства, позволяет клиентам узнавать о компании и ее продукции, не выходя из дома круглосуточно. У клиента появляется возможность в любое время суток воспользоваться web-представительством со своего домашнего компьютера или телефона и как узнать о продукции и компании в целом, так и оформить заказ.

Таким образом, можно сделать вывод, что автоматизация предприятия позволит увеличить его эффективность.

Для анализа деятельности отдела оформления заказов, был использован структурный подход моделирования.

Диаграммы спроектированы при помощи программы Microsoft Visio 2019, которая представляет собой векторный графический редактор, редактор диаграмм и блок-схем [1].

В качестве редактора для создания диаграмм данный программный продукт был выбран вследствие своего удобства, а также обширного списка возможностей, так как в Microsoft Visio возможно создавать различные диаграммы необходимые для моделирования. Еще одним преимуществом является пробный бесплатный период в 30 дней.

Чтобы произвести структурный анализ, была выбрана методология функционального моделирования IDEF0 (I-CAM DEFinition или Integrated DEFinition), которая также является графической нотацией, предназначенной для формализации и описания бизнес-процессов.

Данный стандарт ориентирован на анализ логических отношений

между работами, а не их временную последовательность и представляет организацию как набор модулей [2].

IDEF0 определяет два графических объекта:

- блок, с помощью которого обозначается некоторая функция (действие);
- стрелка, с помощью которой указываются информационные или материальные объекты.

Модель IDEF0 представляет собой набор взаимоувязанных диаграмм. Каждая диаграмма является описанием системы или ее отдельных функциональных блоков в рамках некоторого уровня детализации.

Контекстная диаграмма – диаграмма самого верхнего уровня иерархии, дает самое общее представление о системе.

На контекстной диаграмме (КД) отображается только один функциональный блок (контекстная функция).

Далее контекстная функция декомпозируется на основные функции системы с помощью отдельной диаграммы.

В свою очередь, каждая такая функция может быть разложена на более мелкие и т.д.

При построении модели нужно соотнести каждый функциональный блок более низкого уровня с функциональным блоком более высокого уровня.

1.2.2 Разработка и анализ бизнес модели «как есть»

Перед проектированием системы, следует разработать и произвести анализ бизнес-модели «как есть», что позволит в дальнейшем сформировать недостатки существующей системы и предложить варианты их исправления в модели «как должно быть».

Концептуальная модель, представленная на рисунке 2, отражает основной процесс «Оформление заказа».

Процесс рассматривается с точки зрения клиента, так как целью

является анализ оформления заказа клиентом.

На вход процесс оформления заказа получает определенный ассортимент, установленный компанией и желания клиента. Управляется процесс нормативными документами, уставом компании и законом о защите прав потребителя. На выходе выдается оформленный заказ.

С процессом взаимодействуют:

- менеджер – оформляющий заказ;
- клиент – высказывающий пожелания;
- журнал заказов – бумажный носитель, в который менеджер заносит информацию о заказе.



Рисунок 2 – Концептуальная модель оформления заказа

Для дальнейшего анализа процесса, следует произвести его декомпозицию на меньшие блоки.

Диаграмма декомпозиции основного процесса представлена на рисунке

3 и содержит в себе три подпроцесса:

- «Анализ ассортимента и выбор»;
- «Формирование заказа»;
- «Расчет стоимости и завершение оформления».

Процесс «Анализ ассортимента и выбор» - менеджер представляет покупателю действующий ассортимент и помогает определиться с выбором.

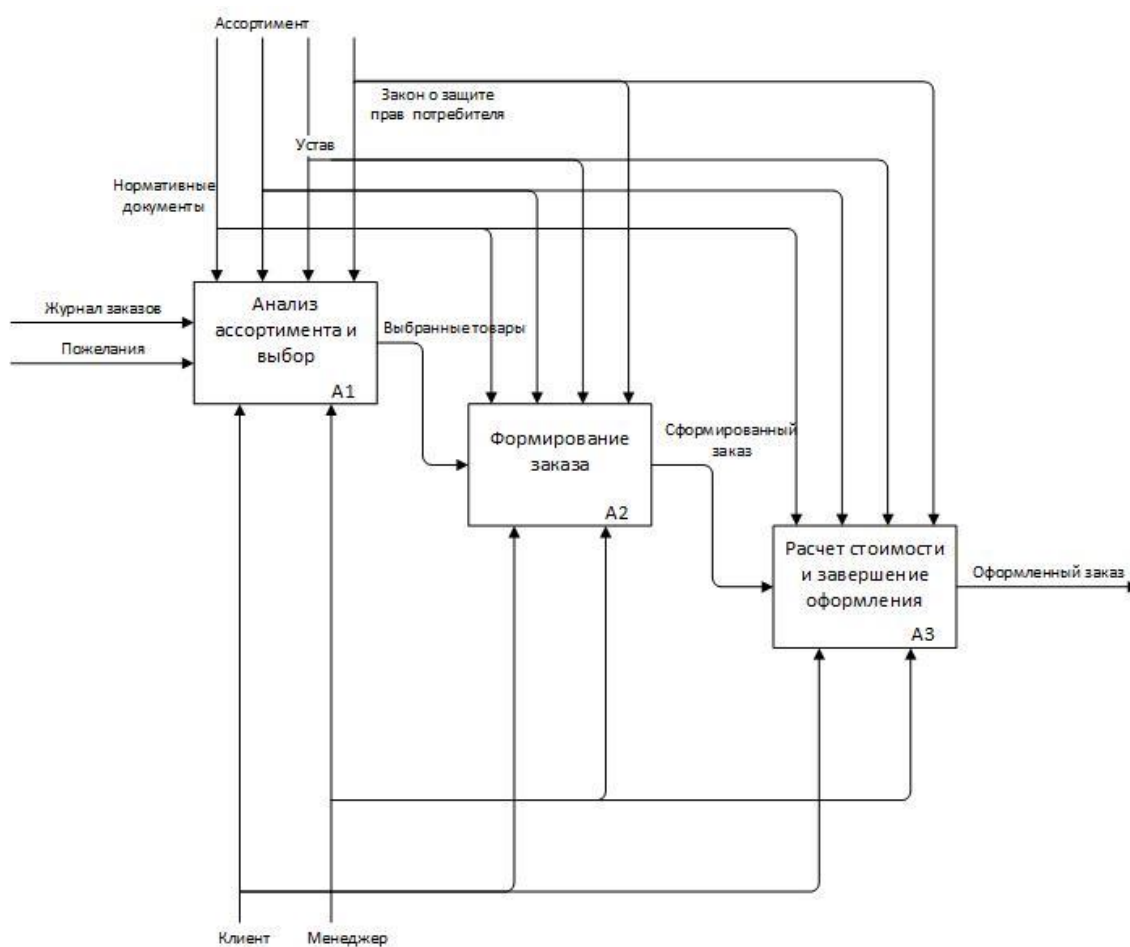


Рисунок 3 – Декомпозиция контекстной диаграммы

Процесс «Формирование заказа» - клиент формулирует окончательную версию своего заказа при помощи менеджера и тот начинает оформления заявки на изготовление.

Процесс «Расчет стоимости и завершение оформления» - менеджер по сформулированному заказу клиента производит расчет и вносит

предложения для более выгодной сделки, а затем предоставляет окончательную стоимость заказа клиенту. После чего вносит подтвержденный заказ в журнал заказов и передает на выполнение.

В результате анализа и построения моделей «как есть» был рассмотрен основной процесс взаимодействия клиента с компанией, а также все его внутренние подпроцессы.

Вследствие этого были выявлены основные недостатки существующей системы:

- отсутствие автоматизации;
- отсутствие веб-сайта с информацией о компании в интернете;
- отсутствие инструмента для поиска и привлечения новых клиентов;
- высокая загруженность менеджера.

Таким образом, можно сделать вывод о том, что метод, используемый сотрудниками компании, устарел и требует модернизации для сокращения трудозатрат и расширения возможностей.

Следовательно, процесс оформления заявки клиентом требует автоматизации, которая может быть осуществлена путем внедрения в систему автоматизированной системы как механизма осуществления процесса, представленной в виде web-представительства.

Целью web-представительства является поиск и привлечение новых пользователей при помощи сети интернет, предоставление информации потенциальным клиентам о компании и продукции, то есть осуществление полного представления компании в сети интернет.

Привлечение новых клиентов без дополнительного инструмента освещения деятельности компании достаточно трудоемкий процесс и практически невозможен без автоматизации.

Внедрение web-представительства позволит:

- экономить ресурсы, необходимые для привлечения новой клиентуры;

- экономить трудозатраты персонала;
- открыть новые пути расширения бизнеса;
- повысить репутацию компании, так как постоянное развитие говорит о компании в хорошем ключе;
- осуществлять прием заказов круглосуточно;
- предоставлять каталог товаров дистанционно.

Для проектирования системы, необходимо сначала провести моделирование описанных выше процессов с учетом внедрения разрабатываемой автоматизированной информационной системы.

1.2.3 Разработка и анализ бизнес модели «как должно быть»

Подробный анализ основного процесса взаимодействия с клиентом, показал, что он нуждается в автоматизации.

Для разработки модели «как должно быть» была выбрана более современная методология проектирования BPMN, так как она больше подходит для проектирования информационных систем.

BPMN (Business Process Management Notation) – это язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией/визуализацией и воплощением бизнес-процесса [22].

Эта нотация представляет описание графических элементов, которые используются при построении схемы бизнес-процесса. Такая схема предназначена для регламентации процесса и его участников, а также производства автоматизации процесса.

BPMN-процесс – это любой бизнес-процесс, отражённый с помощью нотации. Процессы состоят из элементов, каждый из которых обозначается на схеме специальным значком.

Элементы нотации являются элементами графической схемы. Базовые элементы следующие:

- пул и дорожки;
- действия;

- шлюзы или развилки;
- события;
- потоки;
- артефакты.

ВРМН элементы пул и дорожка представляют собой совокупность операций и лиц, выполняющих эти операции, и изображаются в виде прямоугольников.

Пул нужен для обозначения границ, в то время как дорожка для отражения ролей и исполнителей.

ВРМН элемент действие подразумевает по собой единицу работы, которая выполняется в ходе процесса, это могут быть как простые, так и составные действия. Изображаются в виде закругленного прямоугольника.

ВРМН элементы развилка или шлюз являются элементами ветвления и изображаются в виде ромба.

ВРМН элемент событие – это основной элемент ВРМН, который отражает события процесса и изображается в виде круга.

ВРМН элемент поток изображается стрелкой и связывает действия чтобы показать последовательность.

ВРМН элемент артефакт является объектом, не влияющим на бизнес-процесс напрямую, то есть это документы и информация.

Под артефактами в ВРМН понимают объекты, которые не влияют на исполнение бизнес-процесса напрямую. Это могут быть документы, данные, информация.

На рисунке 4 представлена модель основного процесса после его автоматизации в нотации ВРМН.

Автоматизированный процесс теперь не требует личного присутствия клиента при осуществлении заказа, а также снижает нагрузку на менеджера, так как часть полномочий была делегирована web-представительству, а также заменены бумажные носители на электронные, включенные в систему.

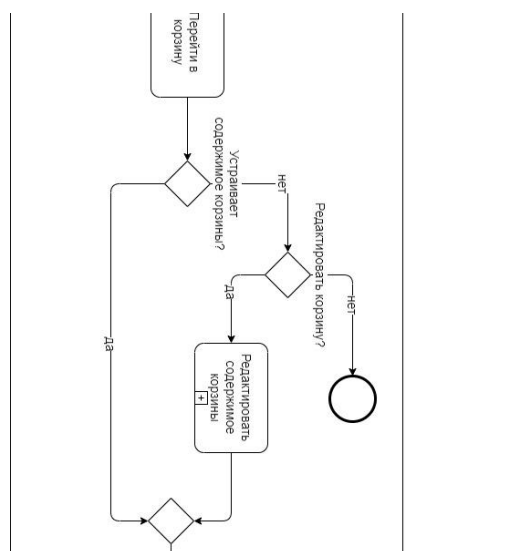


Рисунок 4 – Модель основного процесса «оформление заказа» после автоматизации

Следующим шагом проектирования является моделирование технологического процесса и оформление предметной области в виде модели.

1.2.4 Структурно функциональный анализ основного процесса автоматизации «Оформление заказа»

После проведения анализа процесса «Оформление заказа» следует определение движения потоков данных при помощи диаграмм моделирования информационных потоков DFD.

DFD (Data Flow Diagram) - методология документирования передачи и обработки информации.

Диаграммы DFD обычно строятся для наглядного изображения текущей работы системы документооборота организации и, в частности, могут использоваться в качестве дополнения функциональной модели бизнес-процессов, выполненной в IDEF0 [2].

В нотации DFD модель системы представляется в виде совокупности иерархически упорядоченных и взаимосвязанных диаграмм.

Принципы построения идентичны методологии IDEF0. сначала

строится контекстная диаграмма, а затем производится ее декомпозиция.

Согласно DFD источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те в свою очередь преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям – потребителям информации [3].

Контекстная диаграмма потоков данных компании представлена на рисунке 5.

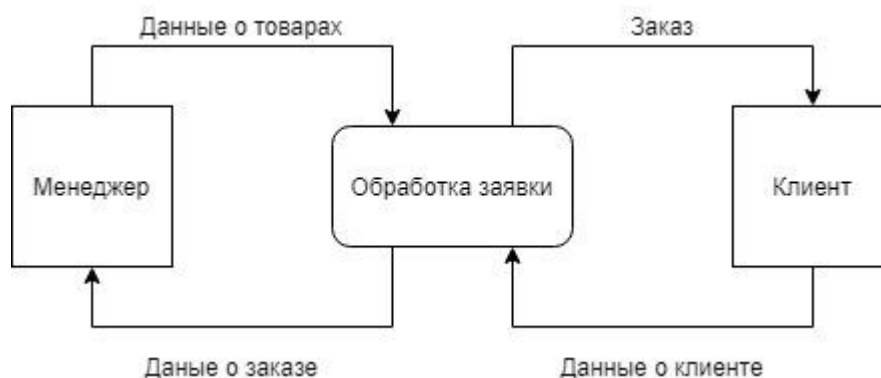


Рисунок 5 - Контекстная диаграмма потоков данных процесса оформления заказа

Благодаря проведению анализа процесса было выделено две внешние сущности: «Менеджер» и «Клиент».

Оформление заказа может быть описано по следующему сценарию:

- от клиента передаются данные в формирование заказа;
- сформированная корзина передается на подсчет стоимости заказа, которая производится при помощи сообщения с хранилищем «Прайс-лист»;
- подсчитанный заказ отправляется в оформление заказа;
- оформленный заказ отправляется в хранилище «Заказ»;

– из хранилища заказ поступает обратно клиенту и менеджеру.

На диаграмме выделены следующие хранилища: «Прайс-лист» и «Заказ», информация о которых будет использована для дальнейшего проектирования системы.

Диаграмма потоков данных позволила произвести описание процесса и потоков информации задействованных в нем. Таким образом, анализ данной диаграммы позволяет продемонстрировать преобразование одних данных в другие и их отношения с различными процессами.

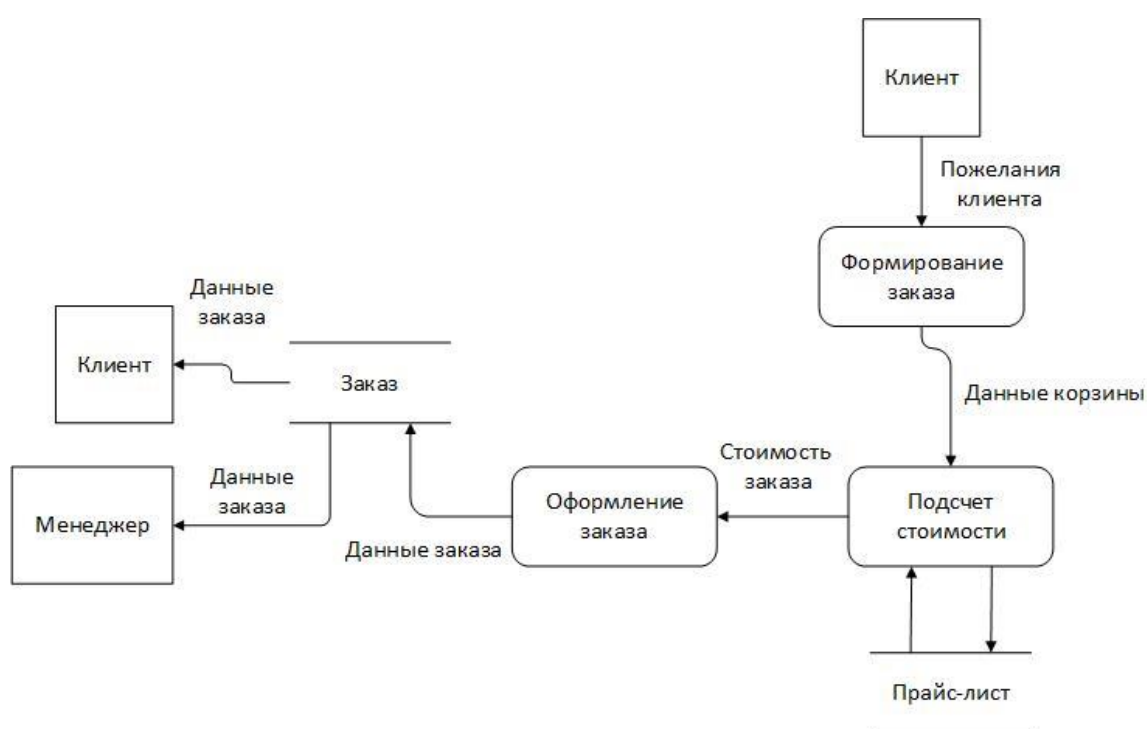


Рисунок 6 – Диаграмма декомпозиции потоков данных

Далее, для осуществления проектирования системы необходимо произвести анализ уже существующих решений на рынке для выяснения требований к программному обеспечению.

Вывод по главе 1

В ходе рассмотрения компании Donutello была проанализирована ее структура, а также основной бизнес-процесс, автоматизация которого является основной целью данной работы.

Моделирование диаграммы информационных потоков будет положено в основу проектирования модели базы данных для будущей системы.

Анализ существующих решений позволил задать планку выполнения критериев для системы и показал возможные сильные стороны будущей системы.

Глава 2. Проектирование web-представительства предприятия

2.1 Обоснование архитектуры проектируемого web-представительства

Автоматизация основного бизнес-процесса «Оформление заказа» является основной задачей данной работы, для решения которой необходимо реализовать и внедрить информационную систему, представленную в виде web-представительства.

Перед реализацией системы, необходимо обозначить требования к ней, путем составления списка этих требований.

Для его составления была выбрана методология FURPS+, которая расшифровывается по первым буквам аббревиатуры: функциональность, удобство использования, надежность, производительность, поддерживаемость, а также дополнительные ограничения если есть.

Разработанные требования к системе представлены в таблице 1.

В качестве архитектуры, для реализации web-представительства, была выбрана трёхзвенная «клиент-серверная» архитектура. Клиент-серверная архитектура определяет общие принципы организации взаимодействия в сети, в которой находятся серверы, узлы-поставщики сервисов и клиенты, которые потребляют функции. В трехзвенной архитектуре сетевое приложение разделяется на две и больше частей, каждая из которых может выполняться на отдельной машине [14].

В модели, разработанной для web-представительства, взаимодействие клиента с web-сервером происходит посредством сетевых протоколов через интернет. Web-сервер состоит из двух компонентов – «Node.js» и «phpMyAdmin».

В свою очередь он взаимодействует с сервером базы данных, который состоит из компонентов «MySQL» и «Базы данных» с использованием SQL-запросов [13].

Таблица 1 – Требования к системе по методологии FURPS+

ID	Описание
Functionality – Функциональные требования	
FUN-01	Предоставление информации о компании
FUN-02	Навигация по сайту
FUN-03	Предоставление прайс-листа
FUN-04	Добавление товаров в корзину
FUN-05	Взаимодействие с пользовательскими запросами
FUN-06	Расчет стоимости выбранных товаров
FUN-07	Регистрация данных заказа
FUN-08	Отправка данных заказа менеджеру и клиенту
Usability - Удобство использования	
USA-01	Основная информация о компании должна быть представлена в виде лендинга
USA-02	Должны быть использованы шрифты Lobster и PT Sans
USA-03	Текст и фон должны соответствовать цветам логотипа
USA-04	Навигация должна отражать структуру web-представительства
USA-05	Блок «Акции» должен быть представлен на слайдере
Reliability - Требования к надежности	
REL-01	Доступ к web-представительству 24 часа в сутки
Performance - Требования к производительности	
PER-01	Время реакции должно быть не более пяти секунд
Supportability - Требования к поддержке	
SUP-01	Время устранения ошибок - час
+ Design - Ограничения проектирования	
DES-01	Должен использоваться язык разметки HTML

Трехзвенная «клиент-серверная» архитектура, в основе которой лежит использование интернет-технологий, показана на рисунке 7.

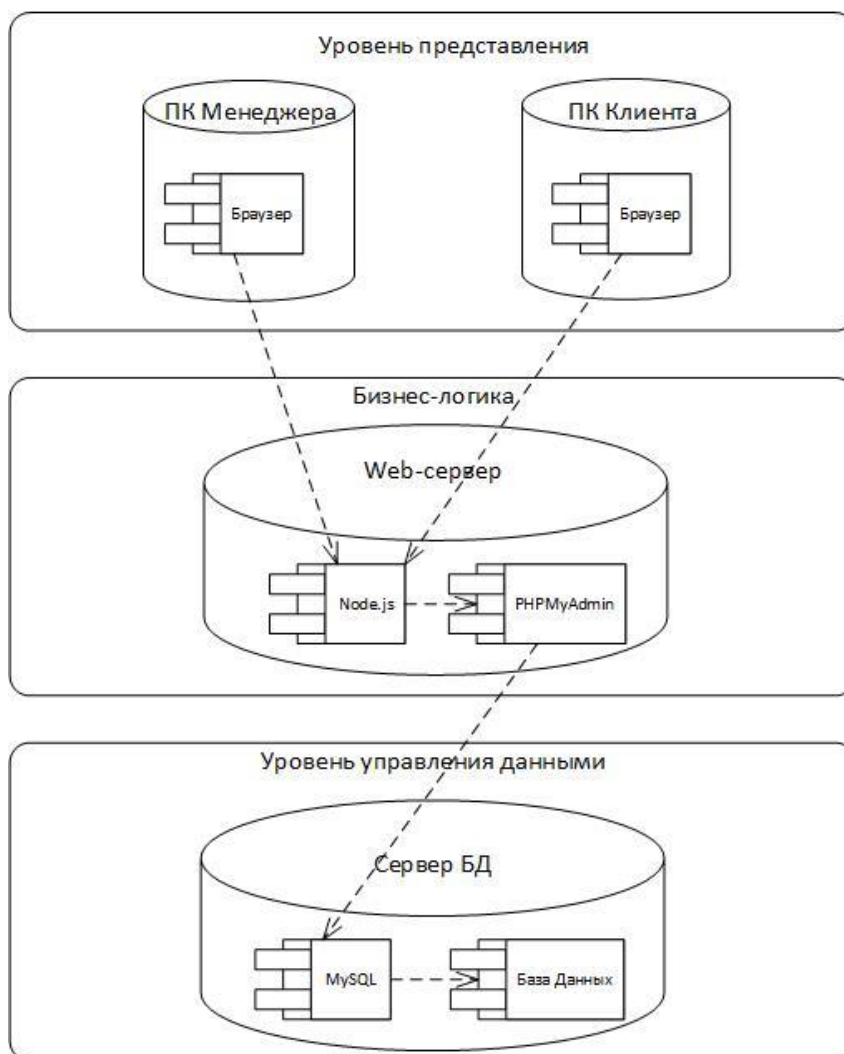


Рисунок 7 – Модель трехзвенной «клиент-серверной» архитектуры

По завершению моделирования архитектуры web-представительства, нужно заняться определением функционального спектра будущей информационной системы.

2.2 Моделирование web-представительства «Donutello»

«Функциональное моделирование — это процесс моделирования функций, выполняемых рассматриваемой информационной системой/объектом, путем создания описательного структурированного графического изображения, показывающего что, как и кем делается в рамках

функционирования объекта и объектов, связывающих эти функции, с учетом имеющейся информации.» [7]

Функциональное моделирование системы будет происходить при помощи языка объектного моделирования UML и программы Microsoft Visio, которая уже была использована для моделирования в предыдущей главе.

UML (Unified Modeling Language) – система обозначений, применяемая для объектно-ориентированного анализа и проектирования [8].

С помощью универсального языка моделирования можно выделить основные функции, которые будут выполняться информационной системой. Функциональное моделирование в первую очередь заключается в построении диаграммы вариантов использования.

Основными элементами диаграммы использования являются участники и прецеденты, отношения между которыми отображают функции системы.

Диаграмма вариантов использования создается для того, чтобы определить контекст будущей системы и сформулировать требования к функционалу.

Перед проектированием самой диаграммы для более точного описания функций и актеров, был составлен список прецедентов с их описанием, представленный в таблице 2.

Таблица 2 – Краткое описание прецедентов

Прецеденты	Описание
Просмотр информации о компании	Ознакомление с информацией о компании, а также ее возможностях
Просмотр цен на продукцию	Ознакомление с информацией о ценах на продукцию, а также акциях
Формирование заказа	Выбор товаров, добавление их в корзину, расчет ориентировочной стоимости заказа и завершение оформления заказа
Подтверждение заказа	Добавление заказа и оповещение о нем менеджера
Редактирование ассортимента	Редактирование ассортимента менеджером

На самой диаграмме было выделено два участника: клиент и менеджер. Диаграмма представлена на рисунке 8.



Рисунок 8 – Диаграмма вариантов использования

Актер «Клиент» имеет четыре прецедента:

- «Формирование заказа»;
- «Просмотр информации о компании»;
- «Просмотр цен на продукцию»;
- «Наполнение корзины».

Актер «Менеджер» имеет два прецедента:

- «Оформление заказа»;
- «Редактирование ассортимента».

Прецеденты «Формирование заказа» и «Оформление заказа» связаны, так как оформленный заказ подтверждается менеджером.

Разработанная модель отображает функции, которые будут присутствовать в разрабатываемой системе.

2.3 Проектирование базы данных web-представительства предприятия

2.3.1 Концептуальное проектирование базы данных web-приложения

Концептуальное проектирование – это первый этап моделирования базы данных, который реализуется при помощи реализации ER-диаграммы.

ER-модель – позволяет описать концептуальные схемы предметной области, при помощи которых в дальнейшем можно будет выделить ключевые сущности и связи этих сущностей [5].

В нотации Питера Чена, которая будет использована для моделирования, есть следующие правила построения:

- независимая сущность отображается на диаграммах прямоугольником с одинарной рамкой, зависимая – с двойной.
- атрибуты, входящие в первичный ключ сущности, выделяются подчеркиванием имени.
- эллипс многозначных атрибутов изображается с двойным контуром, производных – пунктирным.
- связь между сущностями показывается в виде ромба с указанием имени связи внутри него.

Концептуальная ER-модель, построенная на основе методологии Питер Чена, представлена на рисунке 9.

Сущность «Клиент» имеет следующие атрибуты:

- «ID_Клиента» - первичный ключ;
- «Почта»;
- «Номер телефона»;
- «Имя»;
- «Адрес».

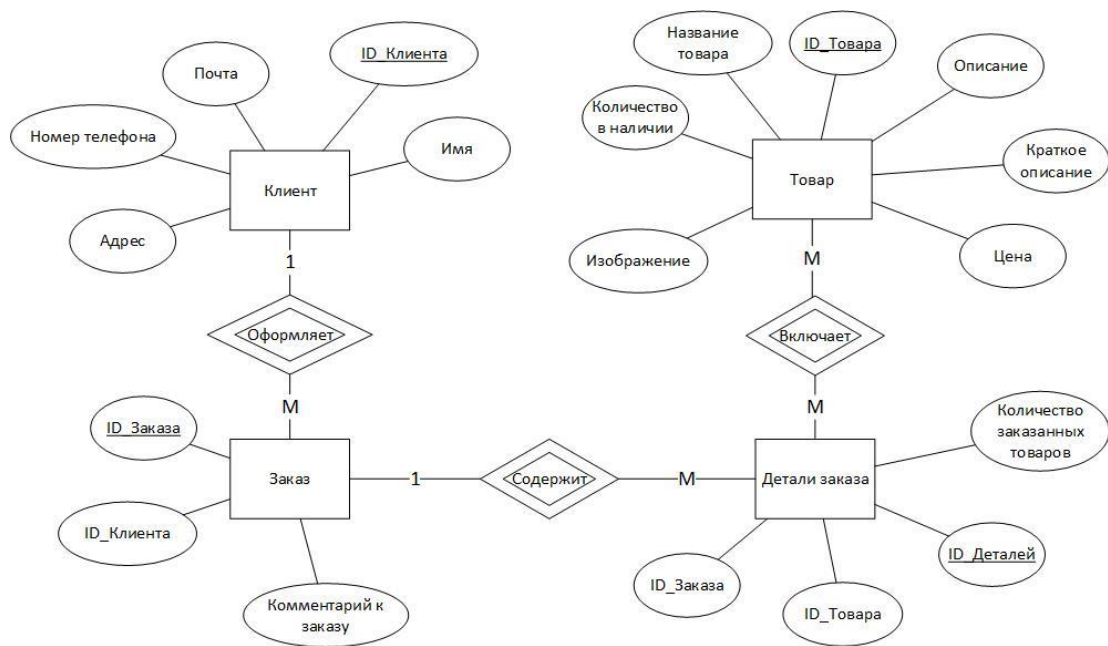


Рисунок 9 – Концептуальная ER-модель, построенная по методологии Питера Чена

«Клиент» связан с сущностью «Заказ» при помощи идентифицирующей связи «Оформляет», имеющей мощность 1:M.

Сущность «Заказ» имеет следующие атрибуты:

- «ID_Заказа» - первичный ключ;
- «ID_Клиента»;
- «Комментарий к заказу».

«Заказ» связан с сущностью «Детали заказа» при помощи идентифицирующей связи «Содержит», имеющей мощность 1:M.

Сущность «Детали заказа» имеет следующие атрибуты:

- «ID_Деталей» - первичный ключ;
- «ID_Заказа»;
- «ID_Товара»;
- «Количество заказанных товаров».

«Детали заказа» связан с сущностью «Товар» при помощи идентифицирующей связи «Включает», имеющей мощность M:M.

Сущность «Товар» имеет следующие атрибуты:

- «ID_Товара» - первичный ключ;
- «Название товара»;
- «Изображение»;
- «Описание»;
- «Краткое описание»;
- «Цена»;
- «Количество в наличии».

После реализации диаграммы сущность-связь следует проектирование логической модели.

2.3.2 Построение логической модели данных web-представительства

Логическая модель данных один из ключевых этапов проектирования информационной системы.

Логическая модель – набор схем отношений, с указанием первичных ключей, а также отношениями внешних ключей.

На основе диаграммы сущность-связь, представленной выше, была спроектирована логическая модель, представленная на рисунке 10.

В логической модели данных было выделено 4 сущности:

- «Клиенты»;
- «Заказы»;
- «Детали заказа»;
- «Продукты».

Сущность «Клиент» имеет атрибут «ID_Клиента», который является первичным ключом. «Клиент» связан с сущностью «Заказы», мощность данной связи 1:М.

Сущность «Заказы» имеет атрибут «ID_Заказа», который является первичным ключом. «Заказы» связан с сущностью «Детали заказа», мощность данной связи 1:М.

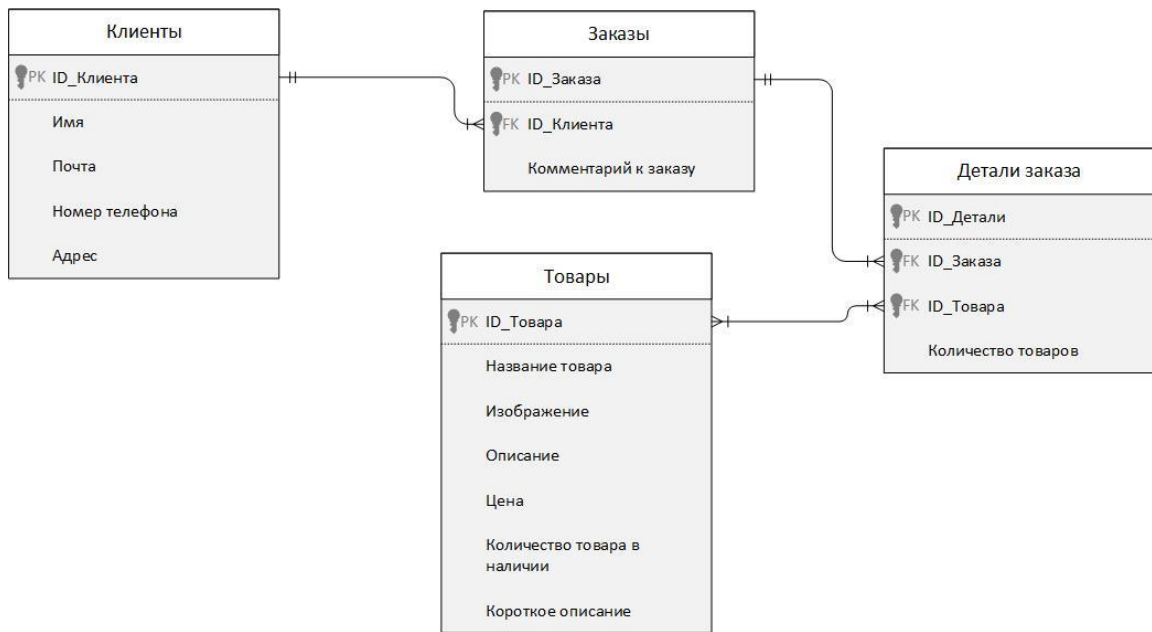


Рисунок 10 – Логическая модель данных

Сущность «Детали заказа» имеет атрибут «ID_Детали заказа», который является первичным ключом. «Детали заказа» связан с сущностью «Товары», мощность данной связи 1:M.

Сущность «Товары» имеет атрибут «ID_Товара», который является первичным ключом.

По завершению логического проектирования, нужно произвести физическое моделирование.

2.4 Физическое моделирование web-представительства предприятия

Последним этапом моделирования информационной системы является физическое проектирование базы данных.

Физическое проектирование определяет реализацию логической модели данных.

«Физическое проектирование базы данных - процесс подготовки описания реализации базы данных на вторичных запоминающих

устройствах; на этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения целостности и средства защиты» [6].

Физическое проектирование является третьим и последним этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных.

Во время предыдущего этапа проектирования была определена логическая структура базы данных.

Физическая модель базы данных разрабатываемой информационной системы представлена на рисунке 11.

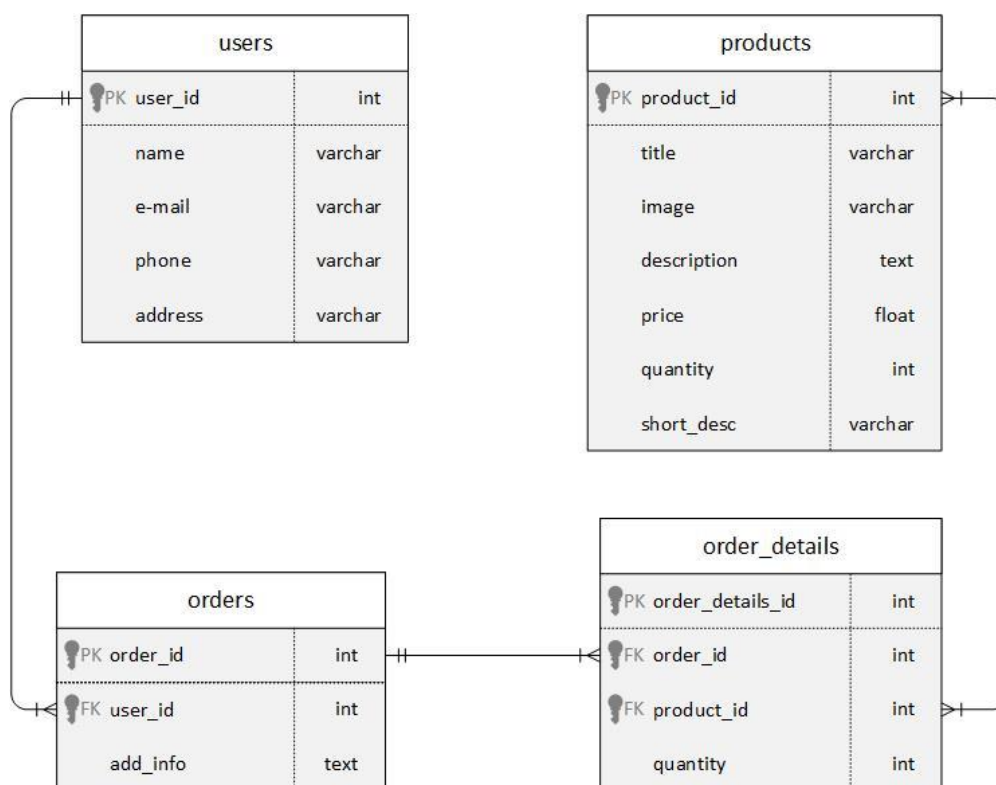


Рисунок 11 – Физическая модель данных

Хотя эта структура не зависит от конкретной целевой СУБД, она создается с учетом выбранной модели хранения данных, например

реляционной, сетевой или иерархической.

Между логическим и физическим проектированием существует постоянная обратная связь, так как решения, принимаемые на этапе физического проектирования с целью повышения производительности системы, способны повлиять на структуру логической модели данных.

Данная диаграмма содержит нужную информацию для реализации СУБД и опирается на все предыдущие модели.

Вывод по главе 2

В данной главе было произведено моделирование будущей информационной системы для компании. В ходе данного моделирования были выделены основные функции web-приложения, спроектированы концептуальная, логическая и физические модели проектируемой системы.

Глава 3 Реализация web-представительства предприятия

3.1 Обоснование выбора технологий реализации web-приложения

Для реализации поставленной задачи, необходимо определить технологии с помощью которых будет производиться разработка.

Web-приложение, архитектура которого представлена выше состоит из трех частей: база данных, серверная часть и клиентская часть.

Так как первая часть — это база данных, первым этапом необходимо определиться с системой управления базами данных (СУБД). Для определения лучшего выбора необходимо установить следующие критерии:

- бесплатность;
- опыт работы с СУБД;
- возможность резервного копирования;
- кроссплатформенность;
- достаточное количество документации;
- распространенность.

В соответствии с данными критериями был проведен сравнительный анализ таких баз данных как – Firebird, MySQL, PostgreSQL, представленный в таблице 3.

Таблица 3 – Сравнительный анализ СУБД

Критерий	MySQL	PostgreSQL	Firebird
Бесплатность	+	+	+
Опыт работы с СУБД	+	-	-
Возможность резервного копирования	+	+	+
Кроссплатформенность	+	+	+
Достаточное количество документации	+	-	+
Распространенность	+	+	-

На основе анализа выбранных СУБД на соответствие поставленным критериям, можно сделать вывод, что наиболее подходящей для использования является MySQL, так как именно она соответствует всем критериям [15, 16].

Помимо этого, для того чтобы упростить работу с MySQL, будет использоваться web-приложение для администрирования СУБД – phpMyAdmin. Оно предоставляет графический интерфейс для базы данных, который позволяет пользователю управлять базой данных без использования запросов.

Backend – вторая часть web-приложения, серверная часть, которая не видна пользователям, к ней относятся: авторизация, хранение и обработка данных, email рассылки и т.д. Для разработки серверной части web-приложения могут быть использованы следующие языки программирования: php и node.js. их сравнительный анализ представлен в таблице 4. [11]

Таблица 4 – Сравнительный анализ php и node.js

Критерий	php	node.js
Простота начала работы	+	-
Помощь и поддержка	+	-
Синтаксис языка	-	+
Инструменты разработчика	-	+
Среда	-	+
Производительность	-	+

Node.js – одна из популярнейших сред выполнения, использующая JavaScript-движок V8 и libuv кроссплатформенную библиотеку поддержки с акцентом на асинхронный ввод-вывод. Она предоставляет возможность писать невероятно производительный серверный код с использованием JavaScript [10, 13].

Frontend – третья часть web-приложения, называется клиентской, потому что заключается в отражении всего что браузер способен вывести на

экран и запустить. Минусом разработки этой части является быстрое развитие трендов – постоянно появляются новые фреймворки и библиотеки для разработки.

Framework (фреймворк) – программная платформа, облегчающая разработку и объединение различных компонентов проекта [18].

Поэтому для выбора фреймворка нужно провести сравнительный анализ, представленный в таблице 5.

Таблица 5 – Сравнительный анализ фреймворков

Критерий	Angular	React	Vue
Является полноценным фреймворком	+	-	+
Достаточное количество документации	+	-	-
Приспособленность к разработке нативных web-приложений	+	-	+
Простое взаимодействие с HTML	+	-	+
Реализация модульности	+	+	-

На основе проведенного анализа, можно сделать вывод, что наиболее подходящим для разработки будет фреймворк Angular.

Выше были рассмотрены основные средства реализации web-приложения, которые и будут применены для разработки данного проекта.

3.2 Разработка базы данных web-приложения

Первым этапом разработки web-приложения выполняется реализация базы данных согласно спроектированным моделям во второй главе.

Для этого сначала необходимо установить все нужное программное обеспечение, такое как Open Server Panel, который представляет собой локальный веб-сервер для Windows.

Данный продукт изначально включает в себя приложение phpMyAdmin, которое упоминалось во второй главе как быстрый способ

создания и настройки базы данных без использования SQL-запросов, что в дальнейшем пригодится менеджеру для формирования базы данных.

В phpMyAdmin создаётся база данных с названием «donutello» аналогично названию компании, для которой производится разработка и названию web-приложения.

Созданная база данных представлена на рисунке 12.

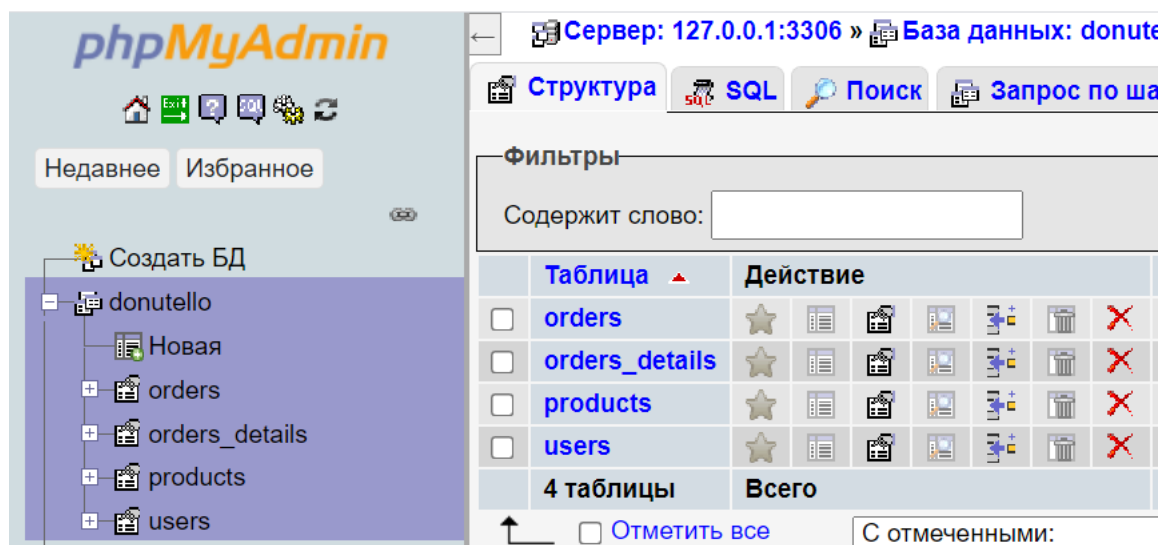


Рисунок 12 – База данных «donutello» в phpMyAdmin

Пример оформленной таблицы «products» представлен на рисунке 13, остальные таблицы создаются по аналогии.

В соответствии со схемой данных создаются четыре таблицы:

- «orders»;
- «order_details»;
- «products»;
- «users».

Сервер: 127.0.0.1:3306 » База данных: donutello » Таблица: products

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(10)			Нет	Нет		AUTO_INCREMENT	
2	title	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
3	image	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			
4	description	text	utf8mb4_unicode_ci		Нет	Нет			
5	price	float			Нет	Нет			
6	quantity	int(10)			Нет	Нет			
7	short_desc	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			

Отметить все С отмеченными:

Печать Анализ структуры таблицы Переместить поля Нормировать

Добавить 1 поле(я) после short_desc Вперёд

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
	PRIMARY	BTREE	Да	Нет	id	0	A	Нет	
	description	FULLTEXT	Нет	Нет	description	0		Нет	

Рисунок 13 – Таблица «products»

Помимо первичных ключей и полей, таблицы связываются отношениями как показано на рисунке 14.

Сервер: 127.0.0.1:3306 » База данных: donutello » Таблица: orders_details

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(10)			Нет	Нет		AUTO_INCREMENT	
2	order_id	int(11)			Нет	Нет			
3	product_id	int(10)			Нет	Нет			
4	quantity	int(11)			Нет	1			

Отметить все С отмеченными:

Печать Анализ структуры таблицы Переместить поля Нормировать

Добавить 1 поле(я) после quantity Вперёд

Индексы

Действие	Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
	PRIMARY	BTREE	Да	Нет	id	0	A	Нет	
	fk_orders_has_products_products1_idx	BTREE	Нет	Нет	product_id	0	A	Нет	
	fk_orders_has_products_orders1_idx	BTREE	Нет	Нет	order_id	0	A	Нет	

Создать индекс для 1 столбца/ов Вперёд

Рисунок 14 – Таблица «order_details»

Поля в таблицах заполняются в соответствии с определенными данными в физической модели, первичные ключи так же.

3.3 Разработка серверной части web-приложения

Для реализации серверной части проекта, в основной папке проекта создается папка с названием «backend», внутри которой настраивается node.js проект с зависимостями к express библиотекам [20].

Структура серверной части проекта по модулям представлена на рисунке 15.

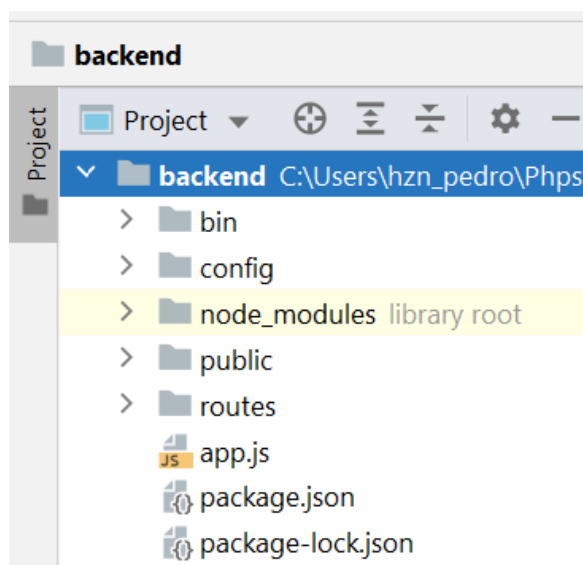


Рисунок 15 – Структура серверной части web-приложения

Папка bin генерируется автоматически средой разработки и включает в себя настройки.

Папка config содержит в себе вспомогательный файл настройки для подключения к базе данных, представленный на рисунке 16.

Внутри данного файла подключается модуль mysql_i, который основан на вторичной инкапсуляции модуля node-mysql, для преобразования синтаксиса SQL в более простой [9].

```
1  const Mysql = require('mysql');
2
3  let connection = new Mysql({ config: {
4      Host: 'localhost', // IP/domain name
5      port: 3306, // port, default 3306
6      user: 'admin', // username
7      passwd: 'admin', // password
8      db: 'donutello'
9  });
10
11  let db = connection.emit({ fromSlave: false, db: '' });
12
13  module.exports = {
14      database: db
15  }
```

Рисунок 16 – Файл helpers.js

После подключения модуля, производится подключение к базе данных.

Папка `node_modules` содержит внутренние библиотеки, установленные во время импорта пакетов.

Папка `routes` содержит два файла: `orders.js` и `products.js`. внутри этих файлов определяется основная логика работы web-сервера.

В оба файла импортируется фреймворк для `node.js` `express`, который может предоставить необходимый функционал.

В файле `orders.js` реализуются основные действия с заказами и маршрутизация.

Таким образом, внутри определяется функция добавления в базу данных нового заказа.

Пример части кода внутри функции представлен на рисунке 17.

В файле `products.js` реализован функционал получения данных из базы данных обо всех товарах или об одном запрашиваемом.

Пример части кода с получением данных из базы данных приведен на рисунке 18.

```

database.table( name: 'orders_details')
  .insert( doc: {
    order_id: newOrderId,
    product_id: p.id,
    quantity: inCart
  }).then(newId => {
database.table( name: 'products')
  .filter({id: p.id})
  .update( doc: {
    quantity: data.quantity
  }).then(successNum => {
  }).catch(err => console.log(err));
}).catch(err => console.log(err));

```

Рисунок 17 – Добавление записи в таблицу order_details

```

database.table( name: 'products as p')
) | .withFields( fields: [
  'p.title as name',
  'p.price',
  'p.quantity',
  'p.description',
  'p.image',
  'p.id'
) | ]
)

```

Рисунок 18 – Запрос информации в базе данных

В файле app.js, который находится в корневой папке определяются все основные характеристики сервера, подключаются необходимые модули и определяются маршруты.

Модули, подключаемые в основном файле описаны в таблице 6.

Таблица 6 – Модули подключаемые в app.js

Модуль	Описание
path	Базовый модуль node.js позволяющий работать с путями к файлам и каталогам
cookie-parser	Модуль для использования cookie
logger	Модуль используемый для логирования
cors	Модуль, который включает в себя механизм разрешения или ограничения запрашиваемых ресурсов на веб-сервере в зависимости от того, где инициирован запрос [19]

После завершения разработки серверной части приложения можно приступить к финальной фазе реализации.

3.4 Разработка клиентской части web-приложения

Последний этап реализации web-представительства — это клиентская часть. Для этой части проекта тоже создается отдельная папка и устанавливаются зависимости с фреймворком Angular [12, 17].

После чего создается структура клиентской части приложения, показанная на рисунке 19.

Папка `node_modules` как и в серверной части содержит в себе библиотеки и файлы импортируемые средой разработки, работа проводится с папкой `src`. В `index.html` как и при работе с обычными `html` файлами прописываются ссылки на скрипты, стили, а также шрифты. Основной функционал моделируется в папке `app`, которая является основным модулем приложения, подключающимся в главном `html` файле. Внутри папки `components` располагаются основные компоненты приложения. Компоненты `header` и `footer` универсальны для всех остальных компонент. Компонент `home` представляет собой главную информационную страницу содержащую всю информацию о компании. Он представлен на рисунке 20.

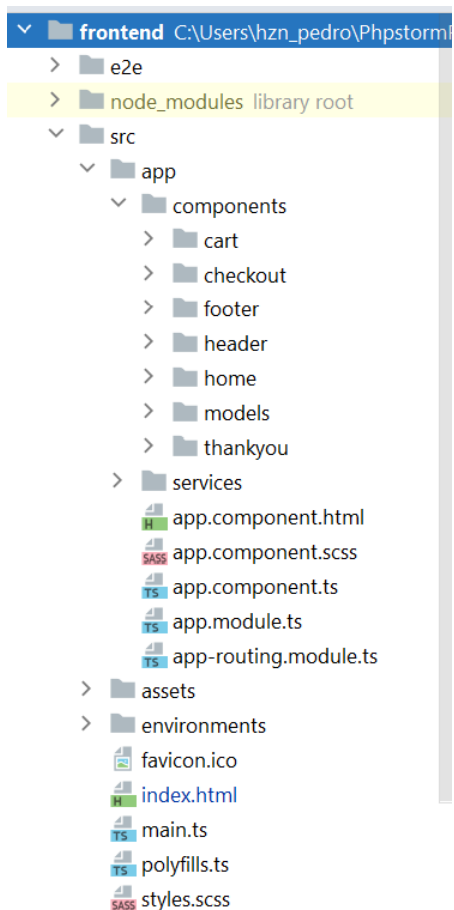


Рисунок 19 – Структура клиентской части web-приложения

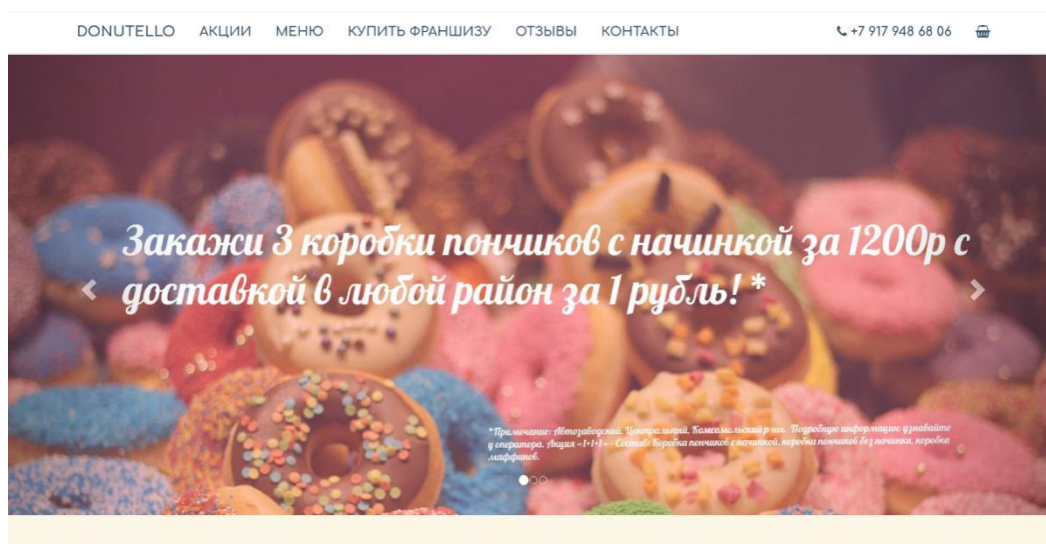


Рисунок 20 – Главная страница

Главная страница состоит из блоков в виде лэндинга, перемещение по которым можно осуществить путем навигационного меню. Блок «Акции» представлен в виде слайдера по запросу заказчика.

Следующий блок «Меню» представлен на рисунке 21 и представляет собой основную функциональность по заказу на сайте.

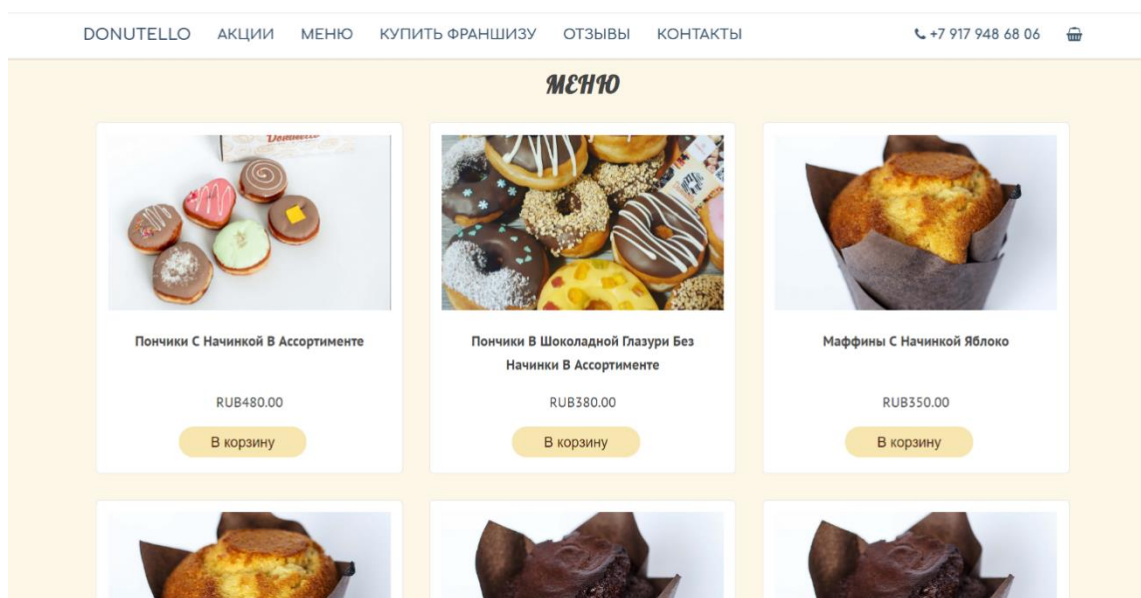


Рисунок 21 – Блок «Меню»

В этом блоке пользователь взаимодействует с корзиной добавляя в нее товары путем нажатия кнопки «В корзину».

Когда товар добавляется в корзину, пользователь видит уведомление об этом и может продолжить просмотр информации о компании дальше.

При нажатии пользователем на изображение товара, открывается скрытый информационный блок о товаре, представленный на рисунке 22.

Данные блока изменяются в соответствии с выбранным товаром, а сам блок скрывается по нажатию на изображение второй раз или же по нажатию на кнопку закрытия внутри самого блока.

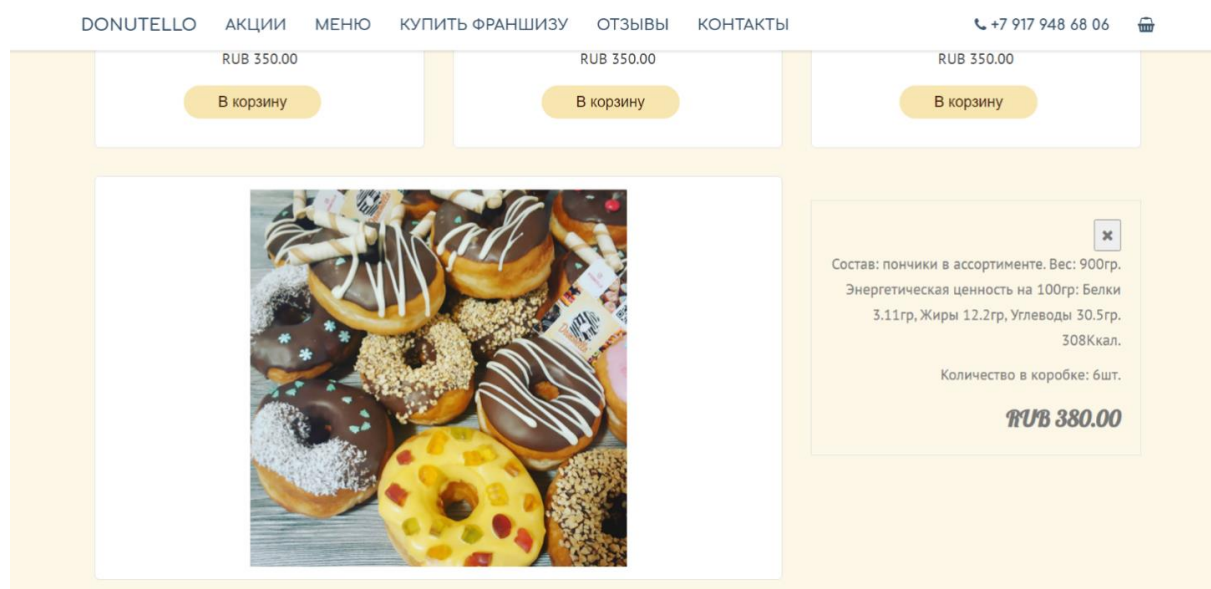


Рисунок 22 – Информационный блок товара

В случае перехода пользователя к корзине, он переключается на следующий компонент cart, который отображает выбранные товары и позволяет редактировать их наличие и количество в корзине, как показано на рисунке 23.

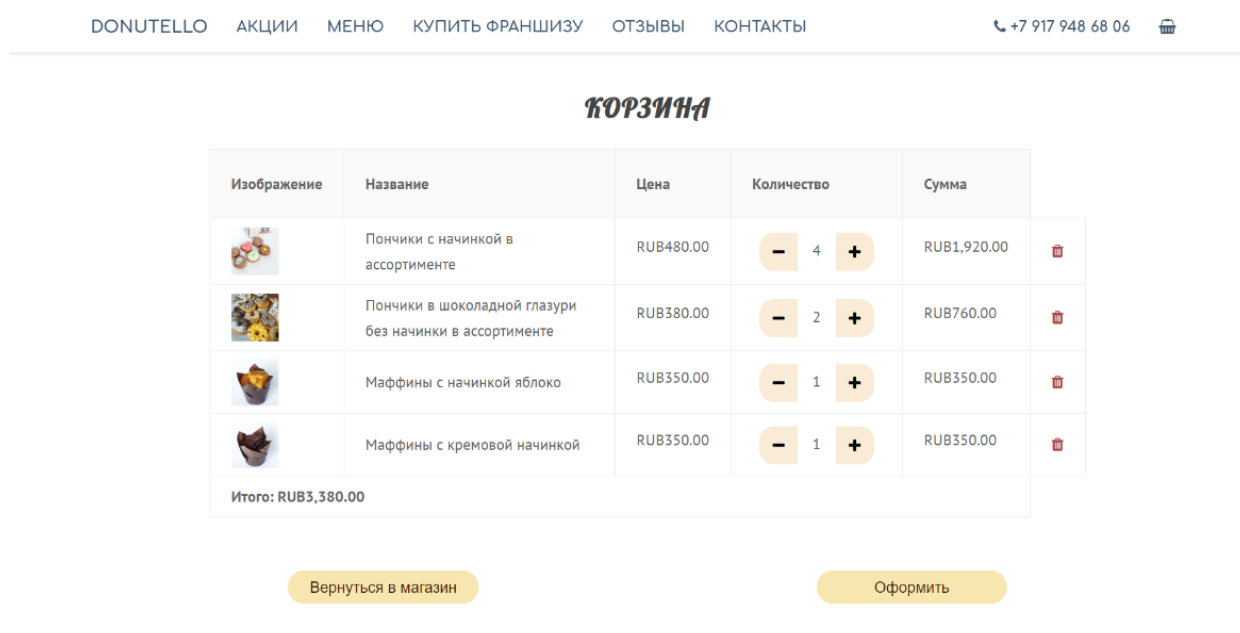


Рисунок 23 – Компонент cart

Содержимое компонента корзины отображается в соответствии с заполненностью корзины. Если она пустая, то пользователь видит сообщение о том, что корзина пуста и кнопку возвращения на главную страницу. Как только заказ отправлен на обработку, корзина становится пустой.

Для реализации всплывающих уведомлений к проекту был подключен модуль toast, пример кода которого представлен на рисунке 24.

```
this.toast.success( message: `${prod.name} добавлен в корзину.`, title: 'Товар добавлен', override: {
  timeout: 1500,
  progressBar: true,
  progressAnimation: 'increasing',
  positionClass: 'toast-top-right'
});
```

Рисунок 24 – Пример кода уведомления об успешном добавлении товара

Клиент может вернуться обратно к просмотру информационной страницы и общему списку товаров.

Другой вариант развития событий, что клиента все устраивает, он может оформить заказ и будет перенаправлен к компоненту checkout, где его попросят указать свои данные для оформления заказа.

Заказ будет зарегистрирован в базе данных, а клиент перенаправлен к компоненту thankyou и получит уведомление о том, что заказ зарегистрирован. Менеджер так же получит уведомление о заказе.

3.5 Тестирование

Для подведения итогов разработки и оценки работоспособности системы, был составлен список тестовых сценариев, основанный на функциональности спроектированной системы, а также требованиях, сформированных для веб-приложения.

Тест-кейс – это сценарий с исходными данными, условиями и

ожидаемым результатом, разработанный для проверки требуемых свойств системы. Также тест-кейсы называются тестовыми сценариями и могут определяться как комплекс входных данных, условий выполнения, процесса тестирования и ожидаемого результата, который нацелен на проверку соответствия требованиям.

Тест-кейсы в количестве 13 штук описаны ниже.

Тест-кейс № 1. Просмотр информации о компании.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Проверить наличие информации о компании.

Ожидаемый результат: Информация присутствует в разных блоках.

Тест-кейс № 2. Навигация по сайту.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Нажать на название блока в навигационном меню.

Шаг 3 – Отображается блок информации в соответствии с пунктом меню.

Ожидаемый результат: Все пункты выполняются.

Тест-кейс № 3. Ознакомление с ценами.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Перейти к блоку «Меню».

Шаг 3 – Проверить наличие цен под товарами.

Ожидаемый результат: Цены отображаются.

Тест-кейс № 4. Добавление в корзину.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Перейти к блоку «Меню».

Шаг 3 – Добавить товар в корзину.

Шаг 5 – Получить оповещение о добавлении товара в корзину.

Шаг 6 – Перейти в корзину и проверить товар.

Ожидаемый результат: Товар в корзине.

Тест-кейс № 5. Изменение количества товара в корзине.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Перейти к блоку «Меню».

Шаг 3 – Добавить товар в корзину.

Шаг 4 – Получить оповещение о добавлении товара в корзину.

Шаг 5 – Перейти в корзину.

Шаг 6 – Увеличить/уменьшить количество товара в корзине.

Ожидаемый результат: Количество товара в корзине изменилось.

Тест-кейс № 6. Удалить товар из корзины.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Перейти к блоку «Меню».

Шаг 3 – Добавить товар в корзину.

Шаг 4 – Получить оповещение о добавлении товара в корзину.

Шаг 5 – Перейти в корзину.

Шаг 6 – Удалить товар из корзины.

Ожидаемый результат: Товар удален.

Тест-кейс № 7. Скачивание документа.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Перейти к блоку «Купить франшизу».

Шаг 3 – Нажать на кнопку «скачать».

Ожидаемый результат: Скачивание началось.

Тест-кейс № 8. Подсчет стоимости товаров в корзине.

Для выполнения теста нужно выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Перейти к блоку «Меню».

Шаг 3 – Добавить товары в корзину.

Шаг 4 – Получить оповещение о добавлении товара в корзину.

Шаг 5 – Перейти в корзину и проверить товары.

Шаг 6 – Проверить правильность вычисления стоимости.

Ожидаемый результат: Стоимость товаров верна.

Тест-кейс № 9. Заполнение формы заказа.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Перейти к блоку «Меню».

Шаг 3 – Добавить товар в корзину.

Шаг 4 – Получить оповещение о добавлении товара в корзину.

Шаг 5 – Перейти в корзину и проверить товар.

Шаг 6 – Перейти к оформлению.

Шаг 7 – Заполнить форму.

Ожидаемый результат: Все поля заполняются.

Тест-кейс № 10. Регистрация заказа.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Перейти к блоку «Меню».

Шаг 3 – Добавить товар в корзину.

Шаг 4 – Получить оповещение о добавлении товара в корзину.

Шаг 5 – Перейти в корзину и проверить товар.

Шаг 6 – Перейти к оформлению.

Шаг 7 – Заполнить форму.

Шаг 8 – Нажать кнопку «Оформить».

Ожидаемый результат: Уведомление что заказ передан менеджеру.

Тест-кейс № 11. Получение уведомления на почту.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Оформить заказ.

Шаг 2 – Зайти на свою почту.

Шаг 3 – Проверить сообщения о заказе.

Ожидаемый результат: Сообщение присутствует.

Тест-кейс № 12. Проверка шрифтов на соответствие PT Sans, Comfortaa и Lobster.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Просмотреть код страницы.

Шаг 3 – Сравнить шрифты указанные в css с PT Sans, Comfortaa и Lobster.

Ожидаемый результат: Шрифты соответствуют.

Тест-кейс № 13. Проверка работы слайдера.

Для прохождения теста необходимо выполнить следующие шаги:

Шаг 1 – Зайти на главную страницу.

Шаг 2 – Переключить кнопки слайдера.

Ожидаемый результат: Слайды меняются.

Таким образом были составлены тестовые сценарии и проведены тесты, результаты которых представлены в таблице 7.

Таблица 7 – Результаты тестирования

№ Тест-кейса	Результат
Тест-кейс № 1. Просмотр информации о компании	Пройден
Тест-кейс № 2. Навигация по сайту	Пройден
Тест-кейс № 3. Ознакомление с ценами	Пройден
Тест-кейс № 4. Добавление в корзину	Пройден
Тест-кейс № 5. Изменение количества товара в корзине	Пройден
Тест-кейс № 6. Удалить товар из корзины	Пройден
Тест-кейс № 7. Скачивание документа	Пройден
Тест-кейс № 8. Подсчет стоимости товаров в корзине	Пройден
Тест-кейс № 9. Заполнение формы заказа	Пройден
Тест-кейс № 10. Регистрация заказа	Пройден
Тест-кейс № 11. Получение уведомления на почту	Пройден
Тест-кейс № 12. Проверка шрифтов на соответствие PT Sans, Comfortaa и Lobster	Пройден
Тест-кейс № 13. Проверка работы слайдера	Пройден

В результате тестирования были проанализированы результаты всех текстовых сценариев. Анализ показал, что тестовые сценарии пройдены успешно.

Тестирование по тест-кейсам выявило, что требования, предъявленные к системе ранее, выполнены.

Вывод по главе 3

В этой главе были выбраны средства реализации программного продукта.

После чего были разработаны: база данных, клиентская часть и серверная часть, которые соответствуют смоделированной системе в предыдущих главах. Также проведено тестирование системы на работоспособность.

Web-представительство было реализовано в соответствии с требованиями и содержит в себе все необходимые функциональные компоненты.

Заключение

В ходе выполнения выпускной квалификационной работы рассмотрено определение понятия web-представительства, выделены основные функции web-представительства.

Дана технико-экономическая характеристика деятельности предприятия.

Проанализирована литература по теме исследования, позволившая провести автоматизацию основного бизнес-процесса предприятия.

Были спроектированы диаграммы IDEF0, BPMN и DFD, которые позволили выявить недостатки существующей системы и построить новую с учетом внедрения web-представительства.

Проведен анализ существующих аналогов, что позволило определить недостатки существующих систем.

Выделены требования по модели классификации требований FURPS+ к проектируемому web-представительству.

Дано обоснование архитектуры проектируемого web-представительства.

При помощи языка UML была составлена диаграмма прецедентов, которая позволила описать систему на концептуальном уровне и определить основные функции необходимые в системе.

Спроектированы логическая и физическая модели данных.

Дано обоснование выбора технологий реализации web-приложения.

В завершении было реализовано web-представительство и проведено его тестирование по набору тестовых сценариев.

В результате тестирования были проанализированы результаты всех текстовых сценариев. Анализ показал, что тестовые сценарии пройдены успешно.

Тестирование по тест-кейсам выявило, что требования, предъявленные к системе выполнены.

Список используемой литературы и используемых источников

1. Microsoft Visio – Краткое руководство [Электронный ресурс] URL: <https://coderlessons.com/tutorials/microsoft-technologies/izuchite-microsoft-visio/microsoft-visio-kratkoe-rukovodstvo> (дата обращения: 19.04.2021)
2. IDEF0. Знакомство с нотацией и пример использования. [Электронный ресурс] URL: <https://trinion.org/articles/idef0-znakomstvo-s-notaciey-i-primer-ispolzovaniya/> (дата обращения: 19.04.2021)
3. DFD – диаграммы потоков данных [Электронный ресурс] URL: <https://infopedia.su/12x891b.html> (дата обращения: 19.04.2021)
4. Фреймворк как программная платформа, Классификация и виды фреймворков (framework) [Электронный ресурс] URL: <https://intellect.icu/frejmwork-kak-programmnaya-platforma-klassifikatsiya-i-vidy-frejmworkov-framework-9515> (дата обращения: 19.04.2021)
5. Нотация Питера Чена [Электронный ресурс] URL: https://studme.org/77222/informatika/notatsiya_pitera_chena (дата обращения: 20.04.2021)
6. Физическое проектирование базы данных [Электронный ресурс] URL: <http://bourabai.kz/dbt/dbms/03.htm> (дата обращения: 20.04.2021)
7. Функциональное моделирование [Электронный ресурс] URL: <http://itstan.ru/funk-strukt-analiz/funkcionalnoe-modelirovanie.html> (дата обращения: 21.04.2021)
8. UML — Краткое руководство [Электронный ресурс] URL: <https://coderlessons.com/tutorials/akademicheskii/uchit-uml/uml-kratkoe-rukovodstvo> (дата обращения: 21.04.2021)
9. Mysqlі – npm [Электронный ресурс] URL: [mysqlі - npm \(npmjs.com\)](https://npmjs.com) (дата обращения: 21.04.2021)
10. Начало работы с Node.js – devSchacht – Medium [Электронный ресурс]: – Режим доступа: <https://medium.com/devschacht/node-hero-chapter-1-239f7afeb1d1> (дата обращения: 25.03.2021)

11. Веб-разработка – с чего начать? / Хабр [Электронный ресурс]: – Режим доступа: <https://habr.com/post/357720/> (дата обращения: 25.03.2021)
12. Yakov Fain and Anton Moiseev Angular Development with Typescript, Second Edition: Manning Publications Co. 2019. 731 p.
13. Thomas Hunter II Distributed Systems with Node.js: O'Reilly Media, Inc., 2021. 377 p.
14. Louis Rosenfeld, Peter Morville, Jorge Arango Information Architecture, 4th Edition: O'Reilly Media, Inc. 2015. 167 p.
15. Mike McGrath SQL in easy steps 4th Edition: In Easy Steps Limited. 2020. 326 p.
16. PARK, DAN SQL for Beginners: The Simplified Guide to Managing, Analyzing Data With SQL: DAN PARK 2020. 183 p.
17. MEAP Edition Manning Early Access Program TypeScript Quickly Version 9: Manning Publications 2019. 514 p.
18. Sufyan bin Uzayr, Nicholas Cloud, Tim Ambler JavaScript Frameworks for Modern Web Development The Essential Frameworks, Libraries, and Tools to Learn Right Now Second Edition: Apress 2019. 591 p.
19. Cors – npm [Электронный ресурс] URL: npmjs.com/package/cors (дата обращения: 21.04.2021)
20. Express – npm [Электронный ресурс] URL: npmjs.com/package/express (дата обращения: 21.04.2021)
21. ВЕБ-представительство компании: сущность и методы создания. [Электронный ресурс] URL: http://www.aselibrary.ru/press_center/journal/irr/irr3648/irr36483710/irr364837103714/irr3648371037143724 (дата обращения: 15.05.2021)
22. Нотация BPMN 2.0: ключевые элементы и описание [Электронный ресурс] URL: <https://www.comindware.com/ru/blog-нотация-bpmn-2-0-элементы-и-описание/> (дата обращения: 15.05.2021)