

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

Разработка и реализация ИС

«Оптовый рынок»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 561 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Маркеева Константина Алексеевича

Научный руководитель  
доцент, к.э.н., доцент

Ю.В. Мельникова

Зав. кафедрой  
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2025

**Введение.** Современный рынок требует от предприятий гибкости, скорости и точности в работе с большими объемами информации. В связи с этим, автоматизация бизнес-процессов становится неотъемлемой частью эффективного функционирования любой организации, особенно в сфере оптовых продаж. Разработка информационной системы (ИС) для автоматизации процессов оптового рынка является важной задачей, которая позволит значительно повысить эффективность работы с клиентами, управление товарными запасами, а также улучшить процессы учета и логистики.

Целью данной дипломной работы является разработка и реализация информационной системы для оптового рынка, которая обеспечит удобное взаимодействие между поставщиками и покупателями и автоматизирует процесс продажи товаров.

Чтобы достичь поставленной цели, необходимо выполнить следующие шаги:

1. Определить ключевые требования к информационной системе.
2. Разработать архитектуру базы данных для системы «Оптовый рынок».
3. Реализовать базу данных и ER-диаграмму с использованием SQL.
4. Создать веб-интерфейс (сайт) для информационной системы.

В работе рассматриваются основные принципы проектирования информационных систем, а также методы и инструменты, которые были использованы для создания системы «Оптовый рынок». Особое внимание уделено созданию пользовательского интерфейса, который будет удобен и интуитивно понятен для конечного пользователя.

Актуальность данной темы заключается в том, что в условиях быстрого развития технологий и глобализации рынка, системы автоматизации становятся важным инструментом для повышения конкурентоспособности и упрощения бизнес-процессов. Реализация эффективной ИС для оптового рынка не только улучшает внутренние процессы предприятия, но и способствует развитию эффективных и удобных способов взаимодействия с клиентами.

**Основная часть.** В первой главе работы проведен анализ современных подходов к проектированию ИС. Рассмотрены ключевые понятия, такие как:

1. Информационная система — совокупность программных, аппаратных и организационных средств для сбора, хранения, обработки и передачи данных.

2. Классификация ИС: транзакционные, управленческие, системы поддержки принятия решений (СППР), CRM-системы.

3. Принципы разработки ИС: модульность, масштабируемость, надежность, безопасность.

Так же внимание уделено веб-интерфейсу как ключевому компоненту ИС. Выделены основные типы веб-сайтов (корпоративные, интернет-магазины, порталы) и этапы их разработки:

1. Планирование требований.
2. Проектирование структуры и дизайна.
3. Программирование (фронтенд и бэкенд).
4. Тестирование и внедрение.

В данном же случае информационная система «Оптовый рынок» разрабатывается как сайт-визитка, направленный на предоставление клиентам информации о товарах и услугах компании, а также возможность оставить заявку на заказ. В отличие от интернет-магазинов, система не предусматривает онлайн-оплаты и корзины, а фокусируется на первичном сборе данных о клиенте с последующим звонком менеджера для уточнения заказа.

Анализ предметной области выявил основные требования к системе:

1. наличие главной страницы с описанием компании и преимуществами;
2. наличие страницы «О компании» с формой обратной связи;
3. наличие страницы «Каталог» с возможностью оформления заявки;
4. наличие страницы «Контакты» с отображением актуальных способов связи и карты;
5. наличие серверной логики для обработки заявок и хранения данных;
6. обеспечение адаптивности интерфейса для различных устройств.

Во второй главе рассмотрено проектирование информационной системы средствами Unified Modeling Language (UML). Первым пунктом данной главы следует, изучение методологии проектирования с использованием UML, в данном пункте была представлена теоретическая информация о UML-диаграммах. Рассмотрены основные типы как поведенческих так и структурных UML-диаграмм.

Основные типы поведенческих UML-диаграмм:

1. Диаграмма последовательности - описывает взаимодействие объектов в рамках конкретного сценария, показывая последовательность сообщений (вызовов методов), которые передаются между объектами.

2. Диаграмма состояний - показывает, как объект изменяет свои состояния в ответ на события, и какие действия выполняются при переходе между состояниями.

3. Диаграмма активности - описывает поток выполнения процессов или операций, включая ветвления, параллельные выполнения и потоки данных. Похожа на блок-схему.

4. Диаграмма коммуникации - показывает взаимодействие объектов, акцентируя внимание на связях между ними и передаваемых сообщениях. Это альтернатива диаграмме последовательности.

5. Диаграмма временных взаимодействий - описывает изменение состояний объектов или взаимодействий во времени, акцентируя внимание на временных ограничениях.

6. Диаграмма обзора взаимодействия - комбинирует элементы диаграмм активности и последовательности, чтобы показать общий поток взаимодействий между объектами.

7. Диаграмма вариантов использования - показывает взаимодействие актеров с системой для анализа требований и обратной связи.

Основные типы структурных UML-диаграмм:

1. Диаграмма классов - описывает структуру классов системы, их атрибуты, методы и отношения между классами. Например, наследование, ассоциация, агрегация, композиция.

2. Диаграмма объектов - показывает конкретные экземпляры классов (объекты) и их связи в определённый момент времени. Это "снимок" системы в конкретный момент.

3. Диаграмма компонентов - описывает физические компоненты системы. Например, модули, библиотеки, исполняемые файлы и их зависимости.

4. Диаграмма развёртывания - показывает, как программные компоненты распределены по физическим устройствам. Например, серверам, компьютерам и как они взаимодействуют.

5. Диаграмма пакетов - описывает группировку элементов системы (классов, компонентов и т.д.) в пакеты, что помогает организовать структуру системы на высоком уровне.

6. Диаграмма композитной структуры - показывает внутреннюю структуру класса или компонента, включая его части и взаимодействия между ними.

Следующие три пункта второй главы описывают диаграммы что использовались при проектировании ИС:

1. Диаграмма вариантов использования. В данной диаграмме определены роли клиента (просмотр каталога, заказ звонка, выбор товара) и администратора (авторизация, внесение изменений в ИС, управление базой данных, обработка заказов).

2. Диаграмма последовательности включает в себя следующие элементы:

- объекты - участники взаимодействия, реальные или абстрактные концепции, например, базы данных;

- временная ось - горизонтальная линия, показывающая порядок действий, выполняемых объектами;

- сообщения - стрелки, демонстрирующие взаимодействие между объектами. Сообщения могут быть синхронными, вызывающий объект ожидает завершения метода перед продолжением работы., или асинхронными, вызов метода не блокирует вызывающий объект, который может продолжать выполнение независимо;

- фреймы - прямоугольные области, предназначенные для группировки объектов и сообщений, что помогает сделать диаграмму более наглядной.

Диаграмма построена для варианта использования «Заказать товар», где Клиент, Главная страница, Сервер и База данных - объекты диаграммы. Детализирован процесс оформления заказа: от выбора товара до сохранения данных в БД.

Рассмотрены основной и альтернативный поток событий, а также поток ошибок.

3. Диаграмма активности предоставляет более детальное представление конкретного сценария использования. В данном случае рассмотрен вариант использования «Заказать товар».

Третья глава посвящена разработке реляционной базы данных на основе MySQL. В данной главе было рассмотрено:

1. Теоретическая информация о базах данных.
2. Обоснован выбор MySQL и выделены его преимущества, такие как:
  - Высокая производительность.
  - Надежность.
  - Кроссплатформенность и совместимость.
  - Простота и удобство.
  - Безопасность

Следом был рассмотрен Node.js как инструмент для реализации серверной части ИС. Были рассмотрены его преимущества:

1. Высокая производительность;
2. Простота интеграции с MySQL;
3. Гибкость и масштабируемость;
4. Большое сообщество и поддержка.

И в заключении была спроектирована ER-диаграмма: выделены сущности «Клиент», «Товар», «Заказ», «Менеджер» и связи между ними (например, один клиент может оформлять несколько заказов).

В четвертой главе описаны этапы создания веб-интерфейса ИС:

В первом пункте главы был рассмотрен выбор среды разработки для реализации веб-интерфейса.

Средой разработки был выбран Visual Studio Code (VS Code) по следующим причинам:

1. Быстродействие и малый размер. В отличие от WebStorm и других тяжелых IDE, VS Code запускается быстро и не потребляет много оперативной памяти.
2. Широкая поддержка языков программирования. Так как разрабатываемая система использует HTML, CSS, JavaScript, Node.js и MySQL, важно иметь поддержку всех этих технологий. VS Code отлично работает с JavaScript и Node.js, предоставляя встроенные функции автодополнения кода, отладку и запуск серверных приложений.
3. Гибкость и расширяемость. В VS Code представлено более 30 000 расширений, которые значительно облегчают работу с кодом.

4. Интеграция с Git. Встроенная поддержка систем контроля версий позволяет отслеживать изменения в проекте и вносить изменения в репозиторий прямо из редактора.

5. Встроенная отладка. Позволяет запускать и тестировать код без необходимости установки дополнительных инструментов.

6. Кроссплатформенность. VS Code одинаково хорошо работает на Windows, macOS и Linux.

После выбора среды разработки было обосновано самостоятельное написание кода для реализации ИС.

Были рассмотрены различные способы создания веб-сайтов: от использования CMS и онлайн-конструкторов до самостоятельного написания кода. Было обосновано решение в пользу ручной разработки, так как она обеспечивает:

1. более высокую производительность;
2. гибкость и масштабируемость;
3. лучшую безопасность данных;
4. полный контроль над функционалом.

Конструкторы сайтов и CMS-системы не позволяют реализовать сложные сценарии взаимодействия с пользователями, ограничения в функционале и слабую защиту от уязвимостей. Кроме того, сайты на конструкторах сложно адаптировать под нестандартные требования и переносить между хостингами.

В третьем пункте рассматривалась реализация веб-интерфейса.

Для начала были перечислены инструменты, которые использовались для разработки проекта:

1. HTML, CSS (включая Flexbox и Grid) – для верстки страниц и адаптивности;
2. JavaScript, jQuery – для взаимодействия с пользователем;
3. Node.js и Express.js – для реализации серверной логики;
4. MySQL – для управления базой данных;
5. Visual Studio Code – в качестве основной среды разработки.

Далее рассматривается процесс написания кода страниц. Веб-интерфейс системы был реализован как адаптивный сайт с четырьмя основными страницами:

Главная страница выполнена с использованием семантической верстки и содержит такие ключевые элементы кода:

- Использование семантической верстки (теги `<header>`, `<section>`, `<footer>` и т. д.);
- Навигационное меню, созданное на основе `<nav>`, адаптированное под мобильные устройства с помощью CSS media queries (набор правил (запросов), которые позволяют адаптировать внешний вид веб-страницы под технические параметры устройства пользователя: ширину и высоту экрана, плотность пикселей, количество поддерживаемых цветов и так далее);
- Большой баннер с призывом к действию.

Адаптивность реализована с помощью media-запросов CSS. «Бургер-меню» появляется при сужении экрана, обеспечивая удобную навигацию с мобильных устройств.

Страница «О компании»

Содержит информацию о команде и преимуществах работы с компанией. Основным элементом является всплывающее окно с формой обратной связи, которое:

1. имеет обязательные поля (имя, телефон, email и др.);
2. проверяет корректность ввода;
3. адаптируется под мобильные устройства;
4. отображает сообщение об успешной отправке данных.

Страница «Каталог»

Содержит список товаров, отображаемых в виде карточек с помощью CSS Grid. Каждая карточка включает:

1. изображение товара;
2. краткое описание;
3. цену;
4. кнопку «Заказать», открывающую ту же форму обратной связи.

Добавление новых товаров не требует перестройки страницы, что обеспечивает масштабируемость решения.



Страница «Контакты» содержит:

1. контактные данные (телефон, почта, мессенджеры);
2. физический адрес;
3. встроенную карту (Google Maps API).

Интерфейс сайта адаптирован под все популярные размеры экранов, включая смартфоны и планшеты, что обеспечивает удобство работы с любого устройства.

**Заключение.** В ходе выполнения бакалаврской работы была проведена разработка и реализация информационной системы «Оптовый рынок» в виде сайта-визитки. В процессе проектирования были изучены основные принципы построения информационных систем, методы моделирования с использованием UML-диаграмм, а также спроектирована и реализована база данных для хранения информации о заказах пользователей.

На основе анализа существующих технологий и инструментов разработки было принято решение использовать Visual Studio Code в качестве среды разработки. Node.js и MySQL для серверной части и управления базой данных. Было обосновано преимущество самостоятельного написания кода перед использованием CMS и конструкторов сайтов, что позволило добиться гибкости, высокой скорости работы и полного контроля над функциональностью системы.

В процессе реализации веб-интерфейса были разработаны основные страницы сайта, обеспечена адаптивность для корректного отображения на различных устройствах, а также внедрено интерактивное всплывающее окно для оформления заявки, что упростило процесс взаимодействия с пользователями. Данные из формы обратной связи обрабатываются и сохраняются в базе данных, после чего менеджер связывается с клиентом для завершения сделки.

Результатом бакалаврской работы стала полностью функционирующая информационная система, отвечающая поставленным целям и задачам. Она позволяет эффективно обрабатывать запросы клиентов, имеет удобный интерфейс и обеспечивает безопасность хранения данных.

Таким образом, поставленные цели и задачи по проектированию и разработке информационной системы «Оптовый рынок» были успешно выполне-

ны. Система обладает необходимой функциональностью, удобством использования и возможностью дальнейшего развития.