

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 информационной системы

«Кондитерский магазин»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Волковой Маргариты Николаевны

Научный руководитель
доцент, к.ф.-м.н., доцент

О.С. Кузнецова

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

Ю.А. Блинков

Саратов 2023

Введение. Современный мир невозможно представить без информационных технологий. Они не только облегчили сферы человеческой деятельности, но и открыли новые горизонты рыночных возможностей.

Активное развитие и совершенствование информационных технологий приводит к усилению конкуренции на рынке. В связи с этим, важно понимать, что предприятиям будет невозможно достичь желаемых результатов своей хозяйственной деятельности без использования информационных систем. Внедрение ИС не только повышает эффективность предприятий, но и позволяет сократить издержки, оптимизировать производственные процессы.

Согласно результатам исследования Data Insight, всемирную сеть используют 82,5% населения России старше 12 лет, и e-Grocery является самым быстрорастущим сегментом внутрироссийского рынка розничной интернет-торговли. К тому же партнер Data Insight в своём докладе «Тренды онлайн-продаж 2022–2023» отметил, что на данный момент происходит перераспределение потребителей из разнообразия «традиционных» магазинов в разнообразие онлайн-магазинов.

Актуальность данной работы связана с увеличением онлайн-продаж на рынке, а также необходимостью внедрения информационной системы для развития и конкурентоспособности организации. Внедрение ИС на основе веб-технологий обеспечит более эффективную работу предприятия за счёт автоматизации основных бизнес-процессов и выхода в мировое информационное пространство сети Интернет.

Целью бакалаврской работы является разработка информационной системы «Кондитерский магазин». Для выполнения поставленной цели в процессе работы необходимо выполнить следующие задачи:

- Изучение теоретических аспектов разработки информационных систем;
- Анализ предметной области;
- Проектирование структуры информационной системы;
- Проектирование и реализация базы данных ИС;
- Разработка и реализация серверной и клиентской части информационной системы.

Структура и содержание бакалаврской работы. Данная работа состоит из следующих разделов:

- Введение;
- Информационная система;
- Проектирование информационной системы «Кондитерский магазин»;
- Проектирование и создание базы данных;
- Реализация информационной системы;
- Заключение;
- Список использованных источников;
- Приложение А Код построения диаграммы вариантов использования;
- Приложение Б Код построения диаграммы последовательности для варианта использования «Оформить заказ»;
- Приложение В Код построения диаграммы активности для варианта использования «Оформить заказ»;
- Приложение Г Исходный код серверной части веб-приложения на Node.js;
- Приложение Д Исходный код реализации страницы продукта.

Первый раздел содержит информацию об информационных системах, их предназначении и о способах построения архитектур ИС, а также включает информацию о предметной области. Раздел состоит из двух подразделов.

В первом подразделе речь идет об основных понятиях проектирования информационной системы.

Под «информационной системой» понимается взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Важно понимать, что ИС предназначена для удовлетворения информационных потребностей потребителей в пределах определенной области. Следовательно информационная система рассчитана на взаимодействие с какими-либо пользователями. ИС должна соответствовать следующим требованиям:

- предоставление клиентских приложений с простым интерфейсом;
- содержать средства и инструменты хранения и доступа к данным;
- регистрация информации в реальном времени;
- открытость и масштабируемость;
- консолидация данных для организационной структуры.

С точки зрения программно-аппаратной реализации можно выделить три основных элемента в общей архитектуре ИС:

- интерфейс пользователя - часть, отвечающая за взаимодействие с конечным пользователем;
- бизнес-логика – совокупность правил, определяющих работу и взаимодействие функций информационной системы;
- информационное хранилище – хранение и обработка данных.

Выделяют два основных подхода построения архитектуры ИС в сети:

- файл-серверная – на файловый сервер хранятся данные, а интерфейс пользователя, бизнес-логика и обработка данных происходит на стороне клиента. Недостатки: при выполнении операций данные каждый раз копируются на сетевой компьютер. В результате увеличивается сетевой трафик;
- клиент-серверная – на сервере размещается информационное хранилище, а интерфейс пользователя, бизнес-логика функционируют на стороне клиента. Недостатки: снижение защиты данных, необходимость использования мощных технических средств связано с функционированием бизнес-логики на стороне клиента.

Для решения этих проблем была предложена трёхуровневая архитектура клиент-сервер, которая базируется на разделении программ, отвечающих за хранение данных, от программ, обрабатывающих их. Этот способ увеличивает сетевой трафик, но зато позволяет более эффективно управлять системой.

В данной работе используется трёхуровневая архитектура клиент-сервер.

Во втором подразделе речь идет об описании предметной области. В работе предметной областью является деятельность кондитерского магазина. Основная задача информационной системы – автоматизировать процесс заказа продукции.

На сайте реализованы следующие страницы, которые доступны всем пользователям:

- «Каталог» – информация о ценах и товарах;
- «Почему мы?» – информация о преимуществах данной компании;
- «Доставка» – информация о доставке и оформлении заказа;
- «Контакты» – информация о контактах;

- «Авторизация» – возможность зайти на сайт;
- «Регистрация» – возможность зарегистрироваться на сайте.

Незарегистрированный пользователь может посещать только вышеперечисленные страницы сайта, а также оформить заказ для одного товара.

Главное преимущества сайта состоит в том, что существует два вида зарегистрированных пользователей: клиент и пользователь с правами администратора. В отличие от обычного зарегистрированного пользователя, который может посещать страницу «Корзина», «Избранное» и «История заказов», у пользователя с правами администратора есть возможность заходить на страницу «Панель администратора», где он может просматривать заказы от клиентов, а также взаимодействовать с БД через сайт (добавлять/удалять товар, тип и начинку, обновлять статус заказа). Пользователь с правами администратора предназначен для работников предприятия.

Второй раздел посвящён проектированию информационной системы с помощью языка UML. Он содержит теоретическую часть, которая предоставляет основную информацию об UML, и практическую, в которой осуществляется проектирование ИС посредством построения трёх диаграмм: диаграммы вариантов использования, диаграммы последовательности, диаграммы активности. Данный раздел состоит из четырёх подразделов.

Первый подраздел содержит основную информацию о языке UML.

UML - общецелевой язык визуального моделирования, который разработан для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес-процессов и других систем.

UML предоставляет пользователю большой набор средств графического описания структуры и поведения сложных систем, который обеспечивает нужный уровень абстракции, способствующий облегчению процесса проектирования.

Структура унифицированного языка моделирования состоит из двух основных элементов:

- Семантика – набор взаимосвязанных логических конструкций: классов, ассоциаций, состояний и других. Семантика определяет содержание модели;
- Нотация языка – визуализация семантики языка.

В языке UML все представления сложных систем фиксируются в виде специальных графических конструкций – диаграмм.

Диаграммы детализируют и конкретизируют различные представления о системе, служат удобным средством передачи информации об особенностях построения ИС.

Во втором подразделе рассматривается диаграмма вариантов использования и осуществляется её построение.

Диаграмма вариантов использования – основная технология моделирования поведения на уровне анализа в UML. Она наглядно представляет возможности системы, которые будут представлены конечному пользователю. При этом механизм функционирования системы не отображается на данной диаграмме.

В диаграмме вариантов использования размещаются элементы - сценарии использования (изображённые в виде овалов) и актеры (изображённые в виде человека).

Связь между элементами в UML происходит с помощью отношений. В диаграмме вариантов использования наиболее часто используются следующие отношения: обобщения, ассоциации, включения и расширения.

Третий подраздел посвящен диаграмме последовательности, описанию потока событий и построению данной диаграммы.

Диаграмма последовательности определяет последовательность сообщений, которыми обмениваются участники. Она описывает взаимодействие системы с действующими лицами в процессе полного или частичного выполнения варианта использования.

В диаграмме последовательности используются два основных элемента - линия жизни и сообщения. Под линией жизни (вертикальная пунктирная линия) подразумевается действующее лицо. Сообщения изображаются в виде горизонтальных стрелок между линиями жизни.

Прежде чем начать строить диаграмму последовательности, необходимо описать поток событий. Он позволяет детализировать сценарий использования, то есть поэтапно описать действия, совершённые конечным пользователем, и поведение моделируемой системы в определённом варианте использования. Каждый вариант использования предполагает наличие нескольких

вариантов поведения системы. В связи с этим поток событий делится на основной и альтернативные.

В четвёртом подразделе речь идет о диаграмме активности, её элементах и особенностях.

Диаграмма активности — это диаграмма, на которой определённая деятельность представлена в виде её составных частей. При этом под деятельностью понимается последовательность действий, которую необходимо совершить для реализации некоторой функциональной возможности информационной системы.

Диаграммы активности состоят из следующих элементов:

- прямоугольники с закругленными углами – действия, которые происходят при варианте использования;
- ромбы – принятие решений по выбору одной из ветвей выполнения процесса;
- стрелки – последовательность выполнения действий;
- чёрный круг – начало процесса;
- чёрный круг с обводкой – окончание процесса.

По своей структуре диаграмма активности напоминает блок-схему. Однако существует важное отличие - параллелизм. На блок-схеме невозможно представить параллельное выполнение, блокирование или ожидание наступления некоторого действия. С другой стороны, в диаграммах активности имеется встроенная поддержка параллелизма. Именно эта особенность делает диаграммы активности столь востребованными для графического представления сценариев.

Все вышеупомянутые диаграммы были построены с помощью веб-сервиса PlantUML Web Server.

Третий раздел посвящён проектированию и созданию базы данных для выбранной информационной системы.

Построение современных информационных систем невозможно без использования технологий базы данных. БД позволяют не только хранить, записывать и обрабатывать данные, но и получать информацию, на основе которой можно решать большое количество разнообразных задач в различных областях деятельности человека.

База данных – система хранения структурированных сведений о предметной области, предназначенных для удовлетворения информационных потребностей пользователей.

Структура любой базы данных определяется моделью данных. Под моделью данных будем понимать совокупность правил создания структур единиц информации и операций над ними в информационной системе. Именно модель данных определяет, как данные будут организованы, сохранены и обработаны в базе данных. Существует множество моделей данных:

- Иерархическая – древообразная структура хранения данных. Она удобна при осуществлении поиска данных;
- Сетевая – структура, построенная на основе графа. При такой модели, любое непродуманное изменение связей может нарушить целостность данных;
- Реляционная – основана на принципе выявления подлежащих описанию в БД сущностей и связей между ними. Из-за борьбы с избыточностью данных в БД приходится разбивать информацию по нескольким таблицам, что со временем может привести к увеличению временных и ресурсных затрат;
- NoSQL – предназначена для хранения неструктурированных (фотографии, видео и др.) и полуструктурированных (JSON, XML) данных.

Для управления данными в БД было разработано специальное программное обеспечение, получившее название системы управления базами данных.

В данной работе была использована реляционная СУБД MySQL. Основным средством общения между конечным пользователем и реляционной БД является язык структурированных запросов (SQL).

SQL – это стандартный язык программирования, применяемый для создания, модификации, поиска и извлечения информации, хранящейся в произвольной реляционной базе данных, управляемой соответствующей СУБД.

ER-моделирование является технологией моделирования данных, которую популяризировал структурный анализ и синтез. Оно и сейчас остаётся популярной технологией для моделирования структур реляционных БД на высших уровнях абстракции.

В ER-диаграмме используются только три элемента:

- сущность – некоторый объект, присутствующий в предметной области;
- атрибуты – набор свойств сущности (пара параметр - значение);
- отношение – отношения между сущностями.

В диаграмме выделяют следующие отношения: «один-к-одному», «один-ко-многим», «многие-ко-многим».

В проектируемой ИС серверная часть веб-приложения реализуется с помощью Node.js. Программная платформа Node.js позволяет выполнять JavaScript-программы без веб-браузера и даёт возможность писать программы практически любого назначения.

Сервер выполняет роль посредника между графическим интерфейсом и СУБД. Node.js приложение взаимодействует с MySQL посредством ORM-системы Sequelize.

ORM – технология программирования, связывающая БД с моделью объектно-ориентированных языков программирования, создавая для элементов СУБД некий виртуальный объектный интерфейс.

В оставшейся части раздела приводится описание кода, отвечающего за созданию базы данных и взаимодействие с ней.

В четвёртом разделе предоставляется информация о библиотеки React, технологии REST API, а также реализуется серверная часть веб-приложения и графический интерфейс.

Node.js — платформа, которая выводит язык JavaScript за пределы браузера и позволяет использовать его в серверных приложениях. Основной упор в Node.js делается на создании высокопроизводительных, хорошо масштабируемых клиентских и серверных приложений.

В данной работе Node.js приложение будет использовать инфраструктуру Express.

Официальный сайт характеризует Express «как минималистичный и гибкий веб-фреймворк для приложений Node.js, предоставляющий обширный набор функций для мобильных и веб-приложений». Express выделяется на фоне других фреймворков за счёт следующий преимуществ в работе:

- Минималистичный – позволяет легко добавлять необходимый функционал в разных его частях;

- Гибкий – механизм действия очень простой: фреймворк принимает HTTP-запрос от клиента и возвращает HTTP-ответ;
- Быстрый – высокая производительность.

При выполнении различных действий, заложенных в бизнес-логику, графический интерфейс посылает HTTP запрос к Node.js по URL. Затем приложение обрабатывает запрос и возвращает HTTP ответ.

Структура Node.js приложения ИС выглядит следующим образом:

- папка `controllers` содержит файлы, отвечающие за взаимодействие с БД;
- папка `error` содержит `ApiError.js`, который осуществляет обработку ошибок со стороны сервера;
- папка `middleware` содержит файлы, отвечающие за промежуточную обработку;
- папка `models` содержит файл `models.js`, описывающий модели данных и взаимосвязь между ними;
- папка `routers` содержит файлы, отвечающие за маршрутизацию;
- папка `static` содержит изображения;
- `.env` – переменные конфигурации;
- `db.js` – передача параметров для подключения к БД;
- `index.js` – основной файл Node.js приложения, отвечающий за запуск.

Графический интерфейс был реализован с помощью библиотеки React.

React – это JavaScript библиотека для создания пользовательских интерфейсов на разных платформах. Она предоставляет мощную модель для работы и помогает создавать декларативные и ориентированные компоненты пользовательского интерфейса. Модель React широко использует функциональные, а также объектно-ориентированные концепции программирования и фокусируется на компонентах как единой основе разработке.

Компоненты представляют собой функции или классы JavaScript, которые получают свойства как входные данные и поддерживают собственное внутреннее состояние. Все, что видит пользователь и с чем взаимодействует, как правило является компонентом React.

DOM – это программный интерфейс, который позволяет JavaScript взаимодействовать с различными типами документов (HTML, XML и SVG). DOM

обеспечивает структурированный способ доступа к документу, его хранения и манипулирования различными его частями.

Одним из основных управляющих элементов технологии является виртуальная DOM. Она представляет собой структуру данных, имитирующую объектную модель документа. Виртуальная DOM не только позволяет эффективно обновлять фактическую DOM, но и обеспечивает отличную производительность.

Стоит отметить, что по умолчанию React поддерживает расширенную версию языка JavaScript под названием JSX. JSX позволяет писать код, похожий на HTML. Использование расширения не является обязательным, но оно значительно облегчает написание и чтение кода.

Передача данных между программами в данной бакалаврской работе осуществляется с помощью технологии REST API.

API – это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.

RESTful API - это прикладной программный интерфейс (API), который использует HTTP-запросы для получения, извлечения, размещения и удаления данных. Он основан на технологии передачи состояния представлений (REST). REST API не сохраняет состояния и существует только для выполнения запросов одного приложения к другому.

Поскольку современные веб-приложения требуют интенсивного взаимодействия клиент-сервер, отправлять данные в свободном формате невозможно. Большинство REST API передают данные в формате JSON.

Благодаря разделению серверной части и графического интерфейса, а также применению REST API, конечные пользователи не имеют доступа к БД, запросы и их обработка скрыты в логике приложения. Это позволяет оптимизировать приложение в сторону высокой производительности и надежности.

В информационной системе «Кондитерский магазин» был реализован следующий функционал пользовательского сайта:

- возможность ознакомиться с каталогом;
- возможность ознакомиться с информацией о преимуществах;
- возможность ознакомиться с информацией о доставке;

- возможность ознакомиться с информацией о контактах;
- возможность оформить заказ;
- возможность авторизоваться/зарегистрироваться на сайте;
- возможность взаимодействовать с корзиной/избранным (для авторизованных пользователей);
- возможность просмотреть историю заказов (для авторизованных пользователей).

Для пользователей с правами администратора был реализован дополнительный функционал:

- возможность взаимодействовать с базой данных: добавлять/удалять товары, начинки и типы (только для пользователя с правами администратора);
- возможность просматривать все заказы и обновлять их статус (только для пользователя с правами администратора).

Заключение. В рамках бакалаврской работы была спроектирована и реализована информационная система «Кондитерский магазин», которая поможет увеличить эффективность работы предприятия за счёт автоматизации основных бизнес-процессов и выхода в мировое информационное пространство сети Интернет.

В ходе работы были приобретены знания в области информационных систем, проанализирована предметная область. С помощью языка UML разработаны и описаны диаграммы вариантов использования, последовательности и активности. Также были приобретены знания в области баз данных и построена ER-диаграмма. На основе полученных знаний с помощью программной платформы Node.js была реализована серверная часть веб-приложения, которая выступала в качестве посредника между графическим интерфейсом и базой данных MySQL.

Затем был изучен современный способ передачи данных с помощью технологии REST API и React. После этого происходила реализация графического интерфейса пользователя на языке программирования JavaScript.

Цели и задачи, поставленные в бакалаврской работе, были достигнуты.

В результате проделанной работы была реализована информационная система для кондитерского магазина.