Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Магнитогорский государственный технический университет   
им Г.И. Носова»

Многопрофильный колледж

Отделение №2 Информационные технологии и программирование

**КУРСОВОЙ ПРОЕКТ**

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

по ПМ.05. Проектирование и разработка информационных систем

МДК.05.02. Разработка кода информационных систем

на тему: Разработка автоматизированной информационной системы деятельности зоомагазина

Исполнитель: Ханнанова Е.З студент 3 курса, группа ИСпВ-21-1

Руководитель: Утралинова С.М, преподаватель МпК

Работа допущена к защите «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Магнитогорск, 2024

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Магнитогорский государственный технический университет   
им. Г. И. Носова»

Многопрофильный колледж

**КУРСОВОЙ ПРОЕКТ**

Тема Разработка автоматизированной информационной системы деятельности зоомагазина

Студент Ханнанова Е.З

Задание

Исходные данные по заданной предметной области:

1. выполнить моделирование информационной системы;
2. спроектировать и разработать базу данных;
3. разработать серверную и клиентскую части приложения;
4. разграничить права доступа;
5. выполнить тестирование информационной системы.

Состав и содержание проекта:

1. построение UML-диаграмм: вариантов использования, последовательности, состояния, деятельности, классов; потоков данных;
2. логическое проектирование, описание концептуальной модели данных, разработка базы данных в выбранной СУБД;
3. разработка интерфейса приложения, логики работы приложения: оформление заявок на услуги|покупку, вывод отчетной документации и т.д.;
4. администрирование приложения, разработка панели администратора;
5. разработка комплекта тестирования информационной системы.

Срок сдачи: « » 20 г.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / С.М Утралинова

« » 2024 г.

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Е.З Ханнанова

« » 2024г.

Магнитогорск, 2024

Содержание

[Введение 4](#_Toc160628397)

[1. Проектирование информационной системы 7](#_Toc160628398)

[1.1. Анализ предметной области 9](#_Toc160628399)

[1.2. Моделирование информационной системы 11](#_Toc160628400)

[1.3. Логическое и физическое проектирование базы данных 20](#_Toc160628401)

[1.4. Выбор и обоснование средств разработки 24](#_Toc160628402)

# Введение

Сейчас 21 век, прошлый век был расценен как повышение информационных и коммуникационных технологий.

В настоящее время все больше развивается сетевые информационные системы для социальных и организационных секторов. Сетевые информационные системы, основанные на веб-технологиях, которые совместно используют различные типы интернет-протоколов и играют важную роль для сбора, хранения и передачи данных из разных источников и мгновенного распространения информации. Скорость и эффективность сетевой информационной системы, которая включает в себя как локальные, так и глобальные сети, базы данных и различные типы программ для обработки информации, облегчает социальную и организационную жизнь людей. Веб-приложения поддерживают операции управления. Такие приложения помогают интегрировать процессы управления качеством, жизненный цикл организационной цели, реализацию и совместное использование производственного плана.

Создание web-приложений, у многих компаний, работающих в сфере цифровых и компьютерных технологий на данный момент считается одним из перспективных направлений деятельности. Перенос бизнес-инструментов и переход от традиционного программного обеспечения на web — это направление, которое необходимо развивать.

Разработка web-приложений становиться все более актуальной темой для многих компаний, ведущих свою деятельность в данной области. Бизнес-организации используют автоматизированную информационную систему (АИС) для развития своего бизнеса и ведения бизнеса во всем мире. Бизнес-организации могут выбирать различные продукты из базы данных. Эта технология позволяет организациям узнать прайс-лист продукта, и они могут заказать различные продукты Веб-приложение. Так что это работает как покупка и продажа СМИ по всему миру. Организации могут отслеживать заказы и обновлять информацию через Интернет. Кроме того, в настоящее время все многонациональные бизнес-организации используют сетевую информационную систему для поддержания связи с другими филиалами. Таким образом, эта технология играет важную роль в развитии бизнеса.

Покупатели идут в Веб-приложение, чтобы найти более низкую цену и купить по максимуму, не выходя из дома, сэкономить время и упростить доставку больших и тяжелых упаковок, которые неудобно транспортировать самостоятельно. Товары для животных чаще оказываются в Веб-приложение-корзине, чем в офлайн: с мая по сентябрь 2020 года доля заказов с такими товарами почти не изменилась – 39% и 38%. Можно говорить о формировании устойчивой модели поведения в Веб-приложение. Тогда как офлайн доля покупок с товарами для животных снизилась в тот же период с 32% до 26%. При этом Веб-приложение-канал характеризуется более редкой частотой покупки, а средняя стоимость покупки в Веб-приложениее практически в два раза выше, чем в офлайне. Также характерно, что цена за килограмм кормов для кошек и собак (как влажных, так и сухих) наиболее высокая в канале зооспециалитета. В первую очередь за счет наличия там профессионального и премиального ассортимента.

Усиление роли Веб-приложение-канала приводит к тому, что традиционные зооспециалитеты превращаются в омниканальных ритейлеров, активно развивают собственное Веб-приложение-направление, работают с маркетплейсами.

Спрос на товары для животных стабильно растет с 2020 По данным исследования NielsenIQ, в 2022 году уже у 72% россиян был питомец. При этом 82% ухаживают за кошками, а 44% — за собаками.

Растущий спрос на зоотовары подтверждают и мультитоварные площадки. Например, в январе — феврале 2023 года на Wildberries продажи товаров для животных выросли на 88%, а на Ozon удвоились по сравнению с аналогичным периодом 2022 года.

Исследования аналитических компаний и опросы показывают, что люди относятся к домашним животным как к членам семьи и склонных их баловать. К примеру, владельцы тратят на своих питомцев в среднем 2000 ₽ в месяц. Основные статьи расходов — корм и ветеринарные препараты. Хозяева также покупают любимцам игрушки, одежду и лакомства. Такое поведение потребителей делает рынок устойчивым и способствует его росту.

Исследования подтверждают: ниша зоотоваров стабильная, в ней нет резких скачков, падений и сезонности — товары одинаково продаются и зимой и летом, потому что питомцам постоянно нужен корм и уход.

Целью данного курсового проекта является разработать автоматизированную информационную систему деятельности зоомагазина.

Исходя из поставленной цели, были сформулированы следующие задачи:

1. Проанализировать предметную область
2. Выбрать технологию разработки
3. Создать UML-диаграмм
4. Спроектировать схему БД
5. Создать БД по выбранной СУБД и разработать запросы
6. Распределить роли
7. Разработать интерфейс клиентского приложения
8. Разработать выходные отчеты
9. Протестировать готовую информационную систему

# Проектирование информационной системы

До недавнего времени термин проект использовался преимущественно в технической сфере, и с ним связывалось представление о разработках сложной документации. В настоящее время любая научно-исследовательская деятельность связана с выполнением различных проектов. Проекты разрабатываются в общественной, политической, международной, технической, театральной, литературной и естественно педагогической сферах деятельности.

Процесс создания проекта называется проектированием. «Проектирование-вид деятельности, дающий начало изменениям в искусственной среде» (Дж.К.Джонс). В широком смысле проектирование — это деятельность по осуществлению изменений в окружающей среде.

Проектирование делится на стадии, этапы и процедуры. Основные этапы создания информационной системы:

1. Предварительный этап

На данном этапе необходимо осознать основные цели и задачи будущего проекта. Для этого ключевые представители Заказчика и Исполнителя организуют встречи, на которых обсуждают концепцию информационной системы, ключевые технические моменты, сроки и объемы выполняемых работ, а также стоимость и источники финансирования. Итогом предварительного этапа, помимо согласованных условий будущего договора, должен стать первый и самый фундаментальный проектный документ – устав проекта.

1. Сбор требований

К этому моменту все ключевые фигуры – участники проекта определены, и ничто не мешает начать собирать и утверждать требования к будущей информационной системе. Представители исполнителя общаются с будущими пользователями и администраторами системы, а также с их руководством. В ходе обследования не только систематизируются требования и пожелания к внедряемому решению, но и анализируется документация, которая должна стать источником исходных данных системы, или формирование которой должно быть в результате автоматизировано.

Результатом данного этапа должно стать появление технического задания на разработку и внедрение информационной системы. Техническое задание должно базироваться на условиях договора и требованиях, изложенных в уставе проекта и содержать следующие разделы (для России структура технического задания регламентируется ГОСТ 34.602 89).

1. Проектирование

На этом этапе усилиями Исполнителя детально проектируются все сценарии, связанные с разработкой и внедрением информационной системы на территории Заказчика. Делается это в соответствии с условиями информационной среды (системного ландшафта) З аказчика и требованиями к интеграции создаваемой системы с уже имеющимися и эксплуатируемыми Заказчиком прочими программными продуктами. Результатом этапа проектирования должно стать оформление следующих разделов технического (концептуального) проекта:

* Архитектура информационной системы.
* Описание структур информационного хранилища (базы данных).
* Проектные решения, представленные детальным описанием сценариев автоматизации всех, затрагиваемых внедрением системы бизнес процессов.
* Сценарии интеграции разрабатываемой информационной системы с внешними программными продуктами.
* Источники исходных данных и варианты первоначального информационного наполнения системы.
* Концепция разграничения прав доступа к данным на основе ролей пользователей, определяющих, в том числе, их полномочия.
* Концепция обучения пользователей информационной системы.

1. Реализация

Этап реализации всех требований к информационной системе, изложенных в Техническом задании и Техническом проекте. В этот период Исполнитель разрабатывает все необходимые программные компоненты, создает структуры базы данных, производит установку, настройку и тестирование всех компонентов информационной системы на своей территории, имитирует сценарии интеграции и т.д. и т.п. Завершение этапа реализации подтверждается появлением таких проектных документов, как руководство по установке и настройке системы, программа и методика испытаний системы, а также шаблон базы данных и реестр всех выполненных программных разработок.

1. Подготовка информационной системы к эксплуатации

Все работы данного этапа уже проводятся на территории Заказчика и включают в себя установку и настройку всех компонентов системы в информационной среде Заказчика, проведение предварительного тестирования, разработку пользовательской документации, обучение пользователей, загрузку исходных данных, проведение испытаний системы в соответствии с программой и методикой испытаний и прочие подготовительные работы.

Информационная система — взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

У внедрения информационных систем в сферу управления и бизнеса много преимуществ.

Внедрение информационных систем может способствовать:

* получению более рациональных вариантов решения управленческих задач за счет внедрения математических методов и интеллектуальных систем и т.д.;
* освобождению работников от рутинной работы за счет ее автоматизации;
* обеспечению достоверности информации;
* замене бумажных носителей данных на магнитные диски или ленты, что приводит к более рациональной организации переработки информации на компьютере и снижению объемов документов на бумаге;
* совершенствованию структуры потоков информации и системы документооборота в фирме;
* уменьшению затрат на производство продуктов и услуг;
* предоставлению потребителям уникальных услуг, отысканию новых рыночных ниш;
* привязке к фирме покупателей и поставщиков за счет предоставления им разных скидок и услуг.

## Анализ предметной области

Предметная область – это конкретная сфера знаний, предметов или явлений, которая исследуется или описывается в рамках определенной дисциплины, профессии или научной области. Она описывает определенный круг понятий, свойств, правил и методов, которые специалисты используют для изучения, анализа и решения задач в данной сфере. Предметная область может быть областью информационных технологий, медицины, экономики, физики, истории и многих других. Она может охватывать как узкую сферу знаний, так и широкий круг определенных предметов или явлений. Например, в области информационных технологий предметной областью может быть разработка программного обеспечения, а в медицине — диагностика и лечение определенных заболеваний.

Предметная область данного курсового проекта – веб-приложение зоомагазин.

Веб-приложения зоомагазины предоставляют ряд преимуществ и удобств, обусловленных современными технологиями и изменением потребительских привычек:

* Удобство покупок (клиенты могут приобрести товары для своих домашних питомцев не выходя из дома, сохраняя свое время и комфорт)
* Широкий ассортимент (веб-приложение может предлагать широкий выбор товаров для различных видов животных, включая корма, аксессуары, игрушки и другие товары, которые могут быть недоступны в обычных магазинах)
* Уникальные услуги (некоторые веб-приложения могут предлагать индивидуальные услуги, такие как доставка домой, подписка на регулярные поставки кормов и средств по уходу)
* Сравнение цен и отзывов (покупатели могут сравнить цены и прочитать отзывы о продуктах, что делает процесс выбора более информированным)
* Доступность круглосуточно (зоомагазины доступны круглосуточно, что позволяет совершать покупки в удобное для клиентов время)

Из-за этих преимуществ и удобств, зоомагазины становятся все более популярными среди владельцев домашних питомцев.

Современные веб-приложения зоомагазинов обычно предлагают широкий спектр функциональности, чтобы обеспечить удобство и удовлетворение потребностей своих клиентов. Вот несколько основных функциональных возможностей:

* Широкий ассортимент продуктов (предлагаются различные товары для домашних животных, включая корма, аксессуары, игрушки, товары для ухода и лечения и многое другое)
* Поиск и фильтрация товаров (клиентам предоставляется возможность быстро находить нужные товары благодаря удобной системе поиска и фильтрации)
* Консультации (некоторые зоомагазины предоставляют возможность получить консультацию от ветеринаров или других специалистов по уходу за животными)
* Персонализированные рекомендации (системы рекомендаций помогают клиентам находить товары, отвечающие потребностям и предпочтениям их питомцев)
* Быстрая доставка (многие зоомагазины предлагают удобные варианты доставки товаров, включая экспресс-доставку или регулярные поставки в соответствии с графиком потребностей клиентов)
* Системы лояльности и скидки (предлагаются программы лояльности, скидки и различные акции для постоянных клиентов)
* Отзывы и рейтинги (клиенты могут делиться отзывами и рейтингами товаров, что помогает другим покупателям принимать обоснованные решения)

В целом, современные веб-приложения зоомагазинов предоставляют разнообразные функциональные возможности, чтобы обеспечить удовлетворение потребностей своих клиентов и обеспечить удобство покупок товаров для домашних питомцев.

А теперь необходимо рассмотреть конкурентов в предметной области:

1. OZON
2. Petshop.ru
3. Wildberries
4. Четыре лапы
5. Бетховен
6. Старая ферма
7. Magizoo.ru
8. Мир корма
9. ZooMag.ru
10. Зоооптторг.рф

Как показывает статистика, основной противник это такие крупные маркетплейсы, как OZON и Wildberries.

## Моделирование информационной системы

В повседневной жизни человек постоянно сталкивается с таблицами: расписание транспорта, турнирная таблица, классный журнал, квитанция для оплаты коммунальных услуг и т.д. Подобные таблицы интуитивно понятны, чтобы в них разобраться не нужны дополнительные инструменты. Все меняется, когда количество столбцов и строк превышает несколько тысяч. С обработкой такого массива данных тяжело справиться даже с помощью электронных таблиц, таких как Excel или «Google Таблицы».

В этом случае большой объем информации распределяется между несколькими таблицами, которые связаны друг с другом. Таким образом формируется база данных. Иными словами, БД – это набор упорядоченной информации, хранящейся в электронном виде.

Есть несколько причин, по которым стоит выбрать именно БД:

* по сравнению с таблицами, базы данных позволяют хранить и обрабатывать гораздо большие объемы сведений;
* база данных работает удаленно, а также имеет систему запросов, что позволяет взаимодействовать с ней сразу нескольким пользователям. Электронные таблицы также поддерживают командную работу, однако системы управления базами данных делают этот процесс более быстрым и безопасным;
* если в базе данных хранится большой объем информации, на скорость работы это никак не повлияет. Электронные таблицы начинают работать медленнее уже после пары сотен строк.

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

Схема UML — концептуальная: это значит, что она оперирует концепциями и связями между ними. Сама диаграмма состоит из фигур, значков, надписей, линий и контуров.

1. Фигуры обычно обозначают ту или иную концепцию: например, объект, класс, группу объектов или что-либо еще. Вариантов фигур в языке множество. Внутри одной фигуры могут находиться другие элементы, главное — чтобы они не пересекали границу.
2. Значки тоже обозначают разные сущности, но отличаются от фигур: внутрь них нельзя ничего поместить. Это могут быть более мелкие атомизированные структурные единицы, а могут быть служебные сущности, например для описаний.
3. Надписи могут быть обычными, подчеркнутыми, курсивными. Они именуют сущности, показывают, что есть что, и могут использоваться для описаний.
4. Линии могут быть прямыми, ломаными, изогнутыми, направленными и ненаправленными, штриховыми и какими-либо еще. Обычно они обозначают связи и зависимости сущностей друг от друга.
5. Контуры — это контейнеры, внутри которых помещаются концепции и связи между ними.

Итак, необходимо приступить к изучению и обзору диаграмм UML. Все UML диаграммы по своей сущности делятся на два вида:

Структурные диаграммы - описывают структуру сложных объектов и систем, показывают статическую структуру системы и ее частей на разных уровнях абстракции и реализации, а также их взаимосвязь

Диаграммы поведения - иллюстрируют взаимодействие с системой и процесс её работы, основное внимание здесь уделяется динамическим аспектам системы программного обеспечения или процесса.

К структурным диаграммам относят следующие 7 типов диаграмм:

* Диаграмма составной структуры
* Диаграмма развертывания
* Диаграмма пакетов
* Диаграмма профилей
* Диаграмма классов
* Диаграмма объектов
* Диаграмма компонентов

А к диаграммам поведения относят следующие типы диаграмм:

* Диаграмма деятельности
* Диаграмма прецедентов (вариантов использования)
* Диаграмма состояний
* Диаграмма последовательности
* Диаграмма коммуникаций
* Диаграмма обзора взаимодействия
* Временная диаграмма

Диаграмма вариантов использования или диаграмма прецедентов (англ. use case diagram) — это графический инструмент универсального языка моделирования (UML), который используется для описания функциональных требований к системе, ее возможных сценариев использования и взаимодействия системы с внешними сущностями (акторами). Данная диаграмма представлена на рисунке 1.

Актеры и варианты использования:

На диаграмме прецедентов варианты использования (Use Case) представляют собой эллипсы, которые отображают функциональные возможности системы или ее части. Они описывают конкретные задачи, которые система должна выполнять для достижения цели, и представляют собой сценарии использования системы с точки зрения ее пользователей.

Актер (Actor) на диаграмме прецедентов представляет внешнюю сущность, которая взаимодействует с системой. Актером может быть любым человеком, группой людей, другой системой или даже другим компонентом системы.

Актеры могут инициировать варианты использования, т.е. задачи, которые система должна выполнить для достижения цели. Например, это может быть пользователь, который хочет купить продукт в интернет-магазине, и вариантом использования — процесс покупки товара.

Отношения (связи):

Отношение «расширение» (extend) — используется, когда один вариант использования может быть расширен другим вариантом использования, если возникают определенные условия. Это отношение показывает, что расширенный вариант использования является необязательным и может быть выполнен только при определенных условиях.

Отношение «включение» (include) — используется, когда один вариант использования использует функциональность другого варианта использования. Это отношение показывает, что один вариант использования является составной частью другого варианта использования.

Пользователь может просмотреть каталог, добавить товар в корзину, сделать запрос на оформление заказа и доставки, иметь личный кабинет, к котором будет отображаться история заказов и товары в избранном.

Админ может редактировать каталог: удалять, добавлять, редактировать карточки товаров, отвечать за составление рейтингов по самым продаваемым товарам, отвечать на вопросы покупателей, вести учет.

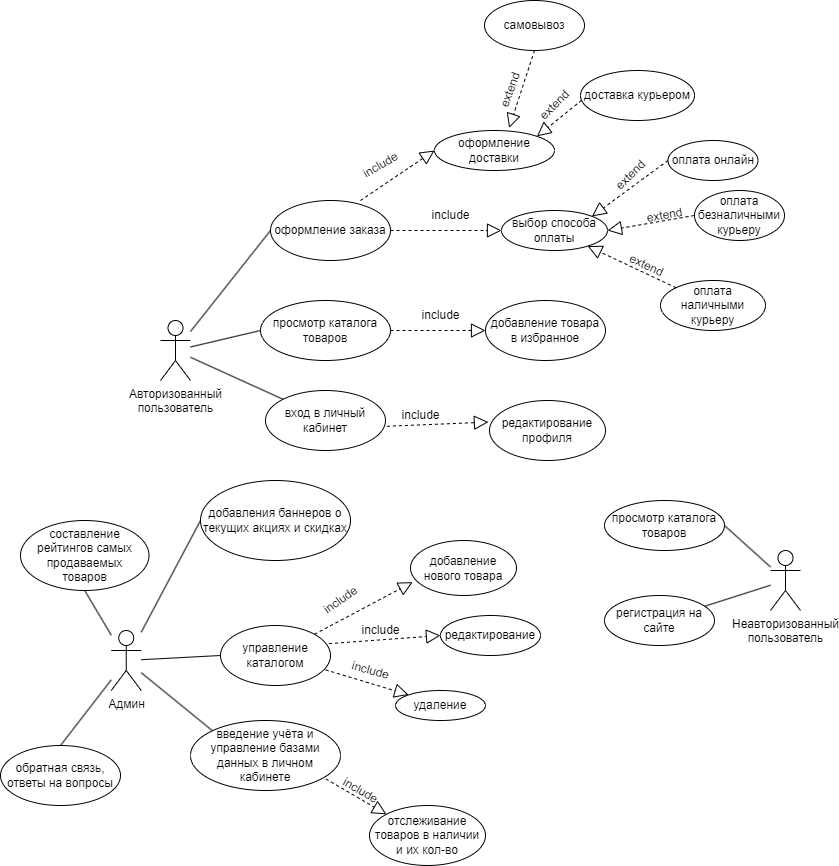
  
Рисунок 1 — Диаграмма вариантов использования

Диаграмма активностей (видов деятельности) - отражает динамические аспекты поведения системы. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой. Данная диаграмма представлена на рисунке 2.

Активности на диаграмме «разбросаны» по беговым дорожкам, каждая из которых соответствует поведению одного из объектов (например, клиента, менеджера, веб-сервера, сервера БД и т.п.). Благодаря этому легко определить, каким из объектов выполняется каждая из активностей. Дорожка - часть области диаграммы деятельности, на которой отображаются только те активности, за которые отвечает конкретный объект. Предназначены дорожки для разбиения диаграммы в соответствии с распределением ответственности за действия. Имя дорожки может означать роль или объект, которому она соответствует.

Пользователь заходит на сайт, после просматривает каталог и выбирает понравившийся товар, добавляя в корзину, перед этим войдя в личный кабинет, после делает заявку на покупку, если товар в наличии происходит оформление и заполнение нужных данных для дальнейшей доставки, если товара не оказалось в наличии пользователь отправляется обратно в каталог для выбора другого товара.

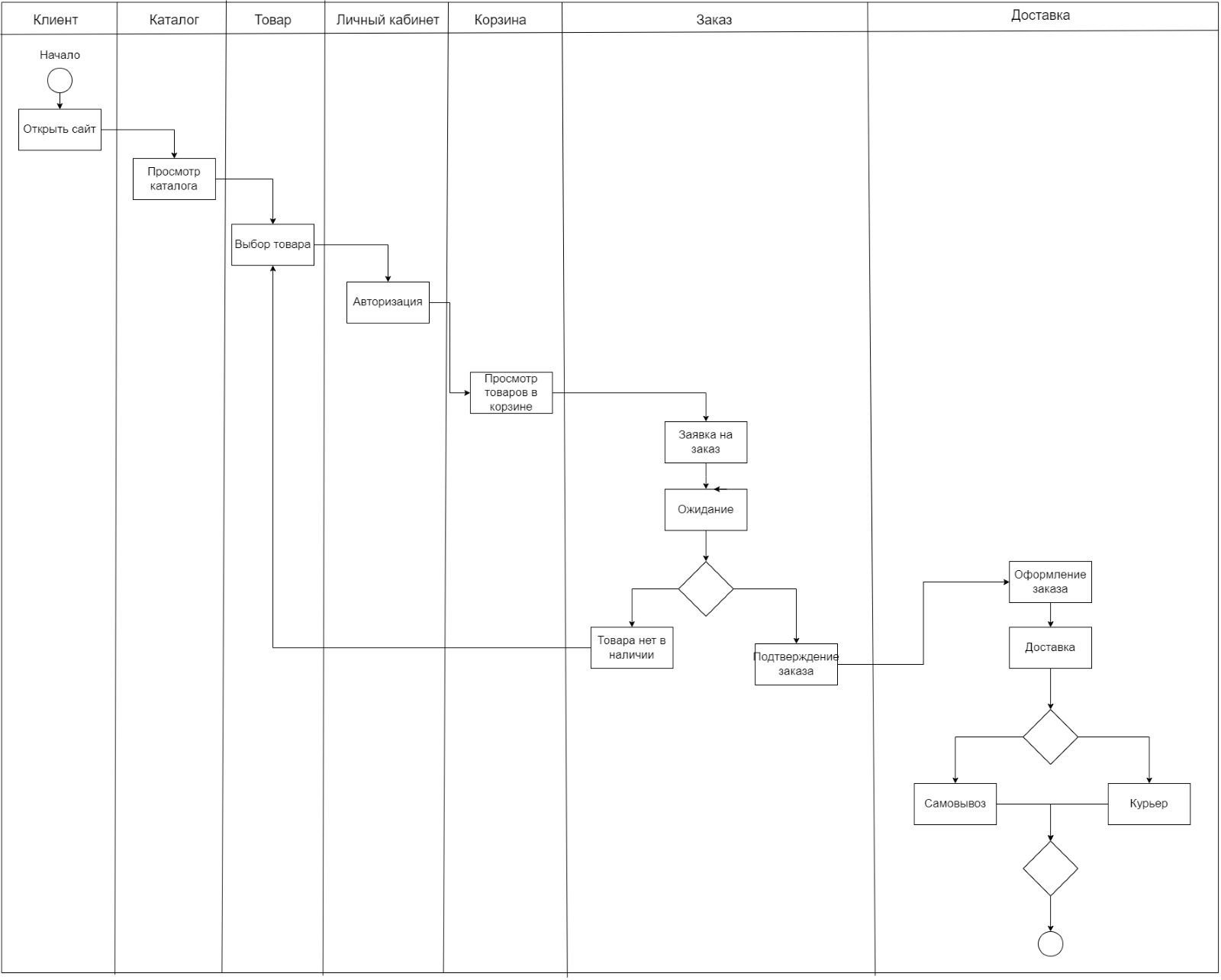
  
Рисунок 2 — Диаграмма деятельности (активностей)

Диаграмма последовательности - это вид диаграммы в UML, который отображает взаимодействие различных объектов в системе во времени. Она позволяет визуализировать последовательность сообщений, которые передаются между объектами в определенном сценарии использования или в определенном временном порядке. Данная диаграмма представлена на рисунке 3.

Основные составляющие диаграммы последовательности включают в себя:

1. Объекты. Объекты, между которыми происходит взаимодействие, представлены на диаграмме последовательности в виде вертикальных линий (иногда называемых "жизненными линиями") с указанием их имени или роли.
2. Сообщения. Сообщения отображают взаимодействие между объектами. Они представляют собой стрелки, которые указывают направление передачи сообщения и содержат информацию о типе сообщения (например, вызов метода, передача данных и т. д.)
3. Фокус управления. Он позволяет указать, в каком объекте в текущий момент сосредоточено управление.

Диаграммы последовательности часто используются для описания поведения системы в ответ на определенные события или для представления конкретных сценариев использования. Они помогают лучше понять порядок взаимодействия объектов и обмен сообщениями в системе.

Пользователь заходит на сайт и просматривает каталог, выбирает понравившийся товар, добавляет в корзину, после делает заявку на покупку, если товар в наличии происходит оформление и заполнение нужных данных для дальнейшей доставки, если товара не оказалось в наличии система отправляет пользователя обратно в каталог для выбора другого товара, при этом товар остается в корзине в неактивном состоянии, до тех пор пока он не появятся в наличии.

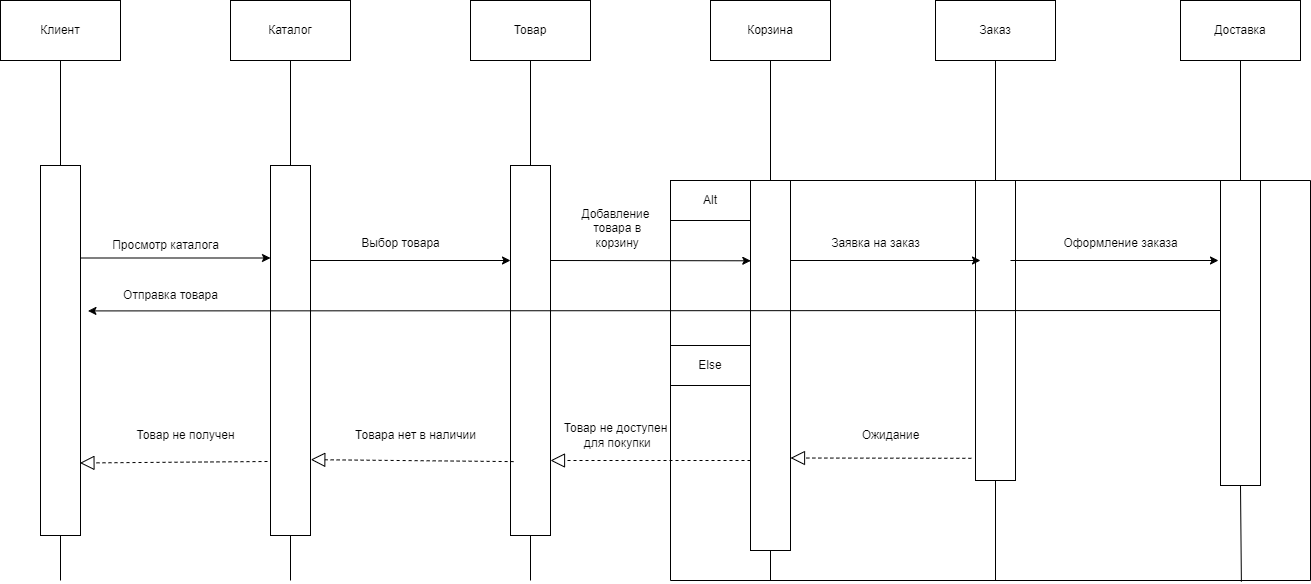


Рисунок 3 — Диаграмма последовательности

Диаграмма состояний позволяет описывать поведение системы.  
В объектно-ориентированном подходе разрабатывается диаграмма состояний единственного класса, демонстрирующая поведение одного объекта в течение его жизни. Диаграмма позволяет проектировать различные способы реакции на события.

Диаграммы состояний — используются для описания поведения сложных систем. Она определяет все возможные состояния, в которых может находиться объект, а также процесс смены состояний объекта в результате некоторых событий. Диаграмма состояний представлена на рисунке 4.

В представленной диаграмме выделены основные компоненты: просмотр, ожидание как следствие просмотра товара, создание заказа, оплата заказа, проверка заказа, отклонение заказа, подтверждение заказа, отправка заказа, ожидание как переход к завершающему состоянию.



Рисунок 4 — Диаграмма состояний

Диаграммы компонентов используются для визуализации организации компонентов системы и зависимостей между ними. Они позволяют получить высокоуровневое представление о компонентах системы. Данная диаграмма представлена на рисунке 5.

Компонентами могут быть: программные компоненты, такие как база данных или пользовательский интерфейс; или аппаратные компоненты, такие как схема, микросхема или устройство; или бизнес-подразделение, такое как поставщик, платежная ведомость или доставка.

Диаграмма компонентов включает в себя следующие основные составные части:

1. Компоненты. (Это основные строительные блоки системы. Компоненты представляют собой фрагменты программного обеспечения, которые могут быть независимо развернуты и заменены. На диаграмме компонентов компоненты изображаются как прямоугольники с названием.)
2. Интерфейсы компонентов. (Компоненты могут взаимодействовать друг с другом через интерфейсы. Интерфейсы демонстрируют, как компоненты связываются и обмениваются данными. На диаграмме компонентов интерфейсы изображаются внутри компонентов с использованием соответствующих обозначений.)
3. Зависимости. (Это связи между различными компонентами, обозначающие, что один компонент зависит от другого. Зависимости могут быть направленными или двунаправленными, и они отображаются на диаграмме стрелками между компонентами.)
4. Интерфейсы зависимостей. (Эти связи демонстрируют, как компоненты используют интерфейсы других компонентов. Они представляются на диаграмме стрелками между интерфейсами компонентов.)

На диаграмме видно, что наша информационная система состоит из следующих компонентов: каталог, товар (карточка товара), личный кабинет, корзина, заказ, база данных, доставка.

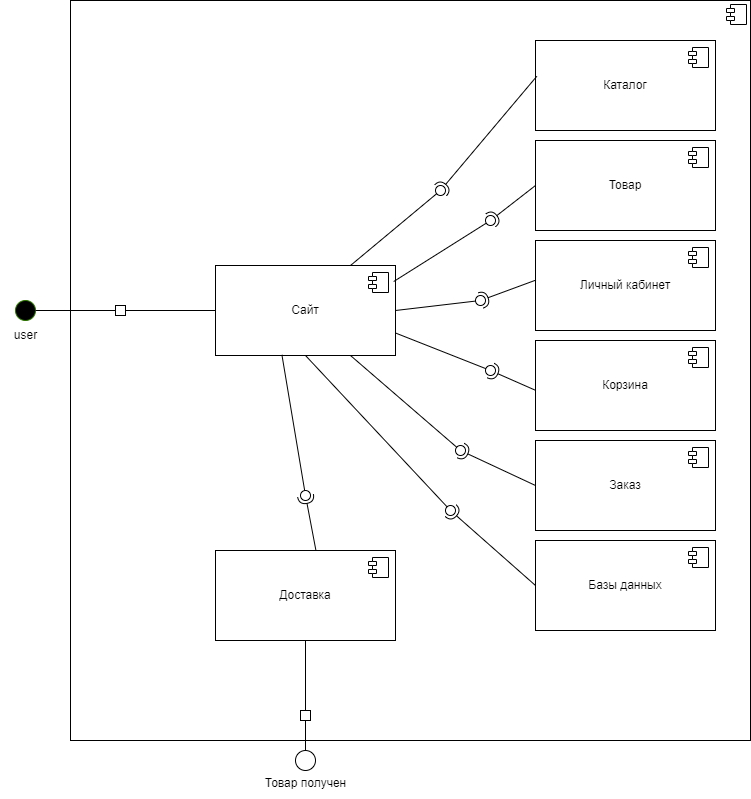


Рисунок 5 — Диаграмма компонентов

## Логическое и физическое проектирование базы данных

Логическая структура базы данных – это описание структуры данных без привязки к конкретной реализации в виде таблиц и столбцов. Она включает в себя описание сущностей (entities), их атрибутов (attributes) и отношений (relationships) между ними. Логическая структура определяет, каким образом данные будут храниться и организовываться, независимо от конкретной технологии базы данных.

Логическая структура реляционной базы данных:

Связи между объектами модели данных реализуются одинаковыми реквизитами — ключами связи в соответствующих таблицах. При этом ключом связи типа 1 : M всегда является уникальный ключ главной таблицы. Ключом связи в подчиненной таблице является либо некоторая часть уникального ключа в ней, либо поле, не входящее в состав первичного ключа (например, код фирмы в таблице СКЛАД). Ключ связи в подчиненной таблице называется внешним ключом.

Логическое проектирование БД является важным этапом в разработке информационных систем и играет ключевую роль в обеспечении эффективного и надежного хранения и обработки данных.

Цель логического проектирования базы данных (БД) – создание структуры БД, которая будет эффективно хранить и обрабатывать данные, соответствовать требованиям бизнеса и обеспечивать целостность и безопасность данных.

Диаграмма ER означает «Диаграмма отношений сущностей», также известная как ERD, — это диаграмма, которая отображает отношения наборов сущностей, хранящихся в базе данных. Другими словами, ER-диаграммы помогают объяснить логическую структуру баз данных. Диаграммы ER создаются на основе трех основных концепций: сущностей, атрибутов и отношений.

Диаграммы ER содержат различные символы, в которых прямоугольники используются для обозначения объектов, овалы для определения атрибутов и ромбовидные формы для обозначения связей.

ER-модель состоит из следующих основных компонентов:

* Сущности – представляют реальные или абстрактные объекты, которые хранятся в базе данных. Каждая сущность имеет уникальный идентификатор (ключ) и набор атрибутов, которые описывают ее свойства.
* Атрибуты – характеристики сущностей, которые описывают их свойства или качества. Атрибуты могут быть простыми (одиночными значениями) или составными (набором значений).
* Отношения – связи между сущностями, которые определяют, как они взаимодействуют друг с другом. Отношения могут быть однонаправленными или двунаправленными.
* Связи – конкретные экземпляры отношений между сущностями. Связи могут иметь атрибуты, которые описывают дополнительные свойства связи.

В модели есть три типа связей:

* «Один-к-одному» — один экземпляр сущности связан только с одним экземпляром другой сущности. Например, пассажир рейса и его место в самолете.
* «Один-ко-многим» — один экземпляр сущности связан со множеством экземпляров другой сущности. Например, у одного пассажира может быть несколько единиц багажа, при этом каждая единица багажа может быть связана только с одним пассажиром.
* «Многие-ко-многим» — множество экземпляров одной сущности связаны со множеством экземпляров другой сущности. Например, аэропорт обслуживает несколько авиакомпаний. При этом каждая авиакомпания может обслуживаться в нескольких аэропортах.

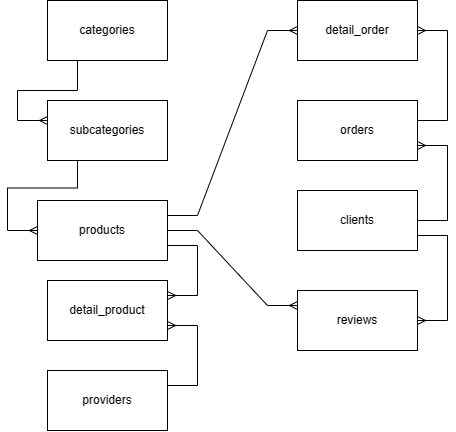


Рисунок 6 — Концептуальная модель базы данных

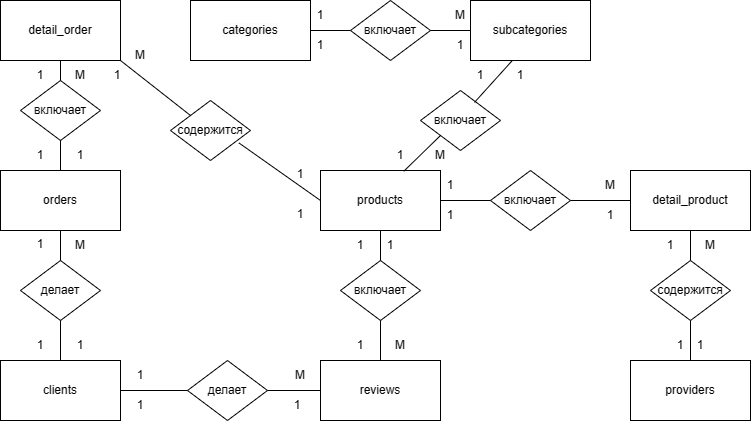


Рисунок 7 — Модель «сущность-связь» базы данных

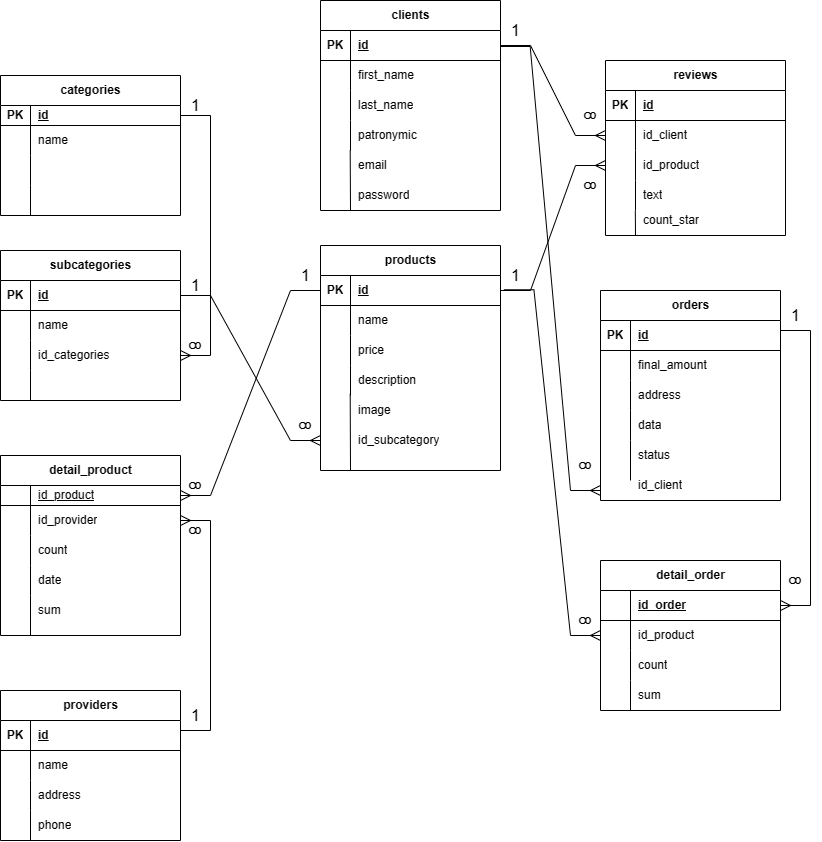


Рисунок 8 — Логическая модель базы данных

База данных для данной ИС будет состоять из 9 таблиц: категории, подкатегории, поставщики, продукты, детали продуктов, отзывы, пользователи (покупатели), заказы, детали заказов:

* таблица категории состоит из следующих полей: id (идентификатор) и name (имя категории);
* таблица подкатегории состоит из следующих полей: id (идентификатор), name (имя подкатегории) и id\_ category (идентификатор категории);
* таблицы поставщики состоит из следующих полей: id (идентификатор), name (имя фирмы, производителя), address (юридический адрес производителя), phone (телефон);
* таблица продукты состоит из следующих полей: id (идентификатор), name(наименование товара), price(цена товара), description(описание, характеристики товара), image(картинка товара), id\_subcategory(идентификатор подкатегории);
* таблица детали продуктов состоит из следующих полей: id\_provider(поставщик), id\_product (продукт), date(дата поставки), count(кол-во), sum(стоимость всей поставки);
* таблица отзывы состоит из следующих полей: id (идентификатор), id\_user (идентификатор пользователя (покупателя)), id\_product (идентификатор товара), count\_star (оценка в звездах от 1 до 5), text (пояснение к оценки);
* таблица пользователи состоит из следующих полей: id (идентификатор), first\_name (имя пользователя), last\_name (фамилия пользователя), patronymic (отчество пользователя если имеется), email (электронная почта), password (пароль от аккаунта);
* таблица заказы состоит из следующих полей: id (идентификатор), id\_user (идентификатор пользователя (покупателя)), final\_amount (общая стоимость заказа), address (адрес доставки), data (дата заказа), status (статус заказа;
* таблица детали заказа состоит из: id\_product (идентификатор товара), id\_order (идентификатор заказа), count(кол-во), sum(сумма).

Физическая модель – логическая модель базы данных, выраженная в терминах языка описания данных конкретной СУБД.

Физическая модель базы данных содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей и т.п.

Физическая модель строится на основе логической с учетом ограничений, накладываемых возможностями выбранной СУБД (в данном случае - MS SQL Server 2012). Имена сущностей и полей (таблиц и столбцов) должны быть записаны латинскими буквами, без пробелов, начиная с буквы. Для каждого атрибута необходимо определить тип данных его значений. Реляционная модель данных требует преобразования связей «многие ко многим» и иерархии наследования и допускает их только на уровне логической модели базы данных.

## Выбор и обоснование средств разработки

Для разработки данной информационной системы использовались следующие фреймворки и языки программирование: Vue.js 3.0, PHP, JavaScript, sql, а также язык гипертекстовой разметки HTML, каскадную таблицу стилей CSS.

Vue.js — это JavaScript-фреймворк модель-представление-представление (MVVM) для создания пользовательских интерфейсов (UI) и одностраничных приложений. Создателем Vue.js является бывший инженер Google Эван Ю, который быстро разочаровался в использовании AngularJS в проектах Google. Он решил извлечь лучшие черты Angular и создать что-то невероятно легкое.

Vue.js — это легкий интерфейсный JavaScript-фреймворк. Он хорошо подходит для любых проектов, в которых задействованы HTML, CSS и JavaScript . И именно поэтому Vue.js в основном используется для создания пользовательских интерфейсов и любых веб-разработок.

PHP – скриптовый или сценарный язык. Это значит, что он предназначен для программ, которые автоматизируют выполнение или интерпретацию мелкой рутинной работы. В противном случае она выполняется вручную.

Поскольку PHP является интерпретатором, его отличают следующие качества:

* Пользователю не приходится освобождать выделенную память.
* Нет необходимости закрывать файлы, когда с ними закончили работать.
* PHP контролирует процесс отладки программ и поиска ошибок.
* Исключает зависание сервиса.

Кроме того, PHP – серверный язык, поэтому все действия выполняются на удаленном веб-сервере.

Это значит, что устанавливать на компьютер PHP не обязательно более того, устройство может вообще не понимать или не знать этот язык, но сайт или приложение будет запущено в стандартном режиме.

Еще одна черта PHP – динамичная типизация. Этот параметр можно охарактеризовать определением типа переменных непосредственно при выполнении программы. А неявные преобразования осуществляются в автоматическом режиме.

JavaScript это язык, который позволяет вам применять сложные вещи на web странице — каждый раз, когда на web странице происходит что-то большее, чем просто её статичное отображение — отображение периодически обновляемого контента, или интерактивных карт, или анимация 2D/3D графики, или прокрутка видео в проигрывателе, и т.д. — можете быть уверены, что скорее всего, не обошлось без JavaScript. Это третий слой слоёного пирога стандартных web технологий, два из которых (HTML и CSS) мы детально раскрыли в других частях учебного пособия.

Популярность JavaScript обусловлена множеством факторов, в том числе большим количеством преимуществ, к которым относятся:

1. Незаменимость при разработке веб сайтов и приложений. Как уже отмечалось выше, JS поддерживается всеми современными браузерами. Причем язык легко интегрируется с версткой и сервером.
2. Высокая скорость работы и производительность. Этот язык позволяет частично обрабатывать веб страницы на стороне пользователя. Это позволяет тратить меньше времени на открытие, а также снижает нагрузку на серверы.
3. Большое количество инструментов и богатая инфраструктура. Если в первое время многие работали исключительно с самим языком, сегодня благодаря наличию множества библиотек, появились удобные и доступные для всех инструменты.
4. Относительная простота. Написание программ занимает обычно меньше времени. При этом, объем кода также обычно меньше, в сравнении со многими другими языками.
5. Широкие возможности для веб-страниц. С помощью JS вы можете оживить любую страницу в сети Интернет. Помимо этого, JS значительно повышает уровень юзабилити приложений и сайтов.
6. Относительная легкость изучения. Начать учиться JS можно даже тем, кто вообще до этого не сталкивался с программированием. Причем визуализация многих действий прибавляет обучающимся энтузиазма.

Что касаемо программного обеспечения, использовались следующие программы: Open Server, PhpMyAdmin, Visual Studio, Figma.

В Figma создавался макет нашего сайта, в Visual Studio писался весь код начиная от верстки и заканчивая работой с сервером, Open Server использовался для запуска и отладки нашего проекта, а в PhpMyAdmin создавалась база данных с нужными таблицами и прочим. Немного о самих программах:

Visual Studio — это IDE (интегрированная среда разработки), очень мощная и масштабная.

С VS Code работают программисты на разных языках. Им активно пользуются веб-разработчики, пишущие на HTML/CSS, JavaScript, PHP. Но редактор поддерживает намного большее количество языков: Python, Go, Ruby, C#, TypeScript и так далее. Он работает и с расширениями и фреймворками для популярных языков — например, с React JS и Vue.js, с языками стилей SCSS и LESS, которые дополняют CSS.

Open Server — это портативный локальный WAMP/WNMP сервер, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов. Представленный пакет программ не является очередной любительской сборкой собранной «на коленке», это первый полноценный профессиональный инструмент, созданный специально для веб-разработчиков.

PhpMyAdmin — это программа для управления базами данных через удобный графический интерфейс.

Её устанавливают на сервер, где хранится сайт, а затем открывают в браузере для удалённого администрирования системы управления базами данных (СУБД).

PhpMyAdmin предназначена для работы с проектами, которые используют веб-сервер Apache, язык PHP и системы MySQL или MariaDB.

Figma — онлайн-редактор, в котором удобно проектировать интерфейсы, создавать макеты сайтов, мобильных приложений, презентации, иллюстрации, логотипы и анимацию. В основном инструментом пользуются дизайнеры, но продакт-менеджерам и разработчикам тоже полезно разбираться в программе.

1. **Разработка информационной системы**

# Весь жизненный цикл информационной системы состоит из следующих этапов:

* Этап 1. Исследование и планирование
* Этап 2. Дизайн и визуальное оформление
* Этап 3. Фронтенд—разработка
* Этап 4. Бэкенд—разработка
* Этап 5. Тестирование и отладка
* Этап 6. Запуск и оптимизация
* Поддержка и развитие

Создание сайта с нуля требует тщательного планирования и последовательного выполнения этапов, чтобы создать функциональный и привлекательный ресурс.

Этап 1:

Самое важное на этом этапе — получить ясное и полное понимание того, каким будет назначение будущего сайта, каких целей нужно достичь с его помощью, а также какова целевая аудитория, которую хочется на него привлечь.

Этап 2:

После планирования можно приступать к проработке вариантов дизайна и визуального оформления. Создание уникального дизайна, отражающего бренд и цель ресурса. Дизайнеры разрабатывают графические элементы, логотипы, иконки, выбирают цветовую палитру и шрифты. Важным аспектом этого этапа является создание макетов страниц и интерфейса пользователя, чтобы визуализировать и оценить внешний вид сайта.

Этап 3:

Фронтенд—разработка — процесс кодирования макетов дизайна с использованием языков разметки HTML и CSS. Разработчики создают структуру, вставляют графические элементы и настраивают визуальное оформление.

Этап 4:

Бэкенд—разработка — этап, на котором создается серверная логика и база данных, которые обеспечивают функциональность сайта. Разработчики выбирают язык программирования, такой как PHP, Python или Ruby, и фреймворк для более эффективной работы. Они реализуют функциональность: обработку форм, авторизацию пользователей, взаимодействие с базой данных.

Бэкенд—разработка также включает тестирование и отладку функциональности, чтобы убедиться в ее правильной работе.

Этап 5:

Разработчики проводят тестирование и отладку, чтобы проверить работоспособность и корректность функциональности нового ресурса. Они выполняют проверку на различных устройствах и браузерах, чтобы убедиться, что сайт выглядит и функционирует одинаково хорошо во всех условиях. Если возникают ошибки или неполадки, разработчики их исправляют и проводят дополнительные тесты.

Этап 6:

После перепроверки сайта, он может быть загружен на сервер. Обычно для этого используется FTP-клиент. После загрузки сайта на сервер, необходимо провести еще один тест для того, чтобы быть уверенным, что во время загрузки не произошло непредвиденных ошибок и все файлы целы и невредимы. Программисты загружают файлы сайта на выбранный хостинг—провайдер и настраивают соединение с базой данных (если используется). Важно выбрать надежного хостинг—провайдера, который обеспечит стабильное функционирование и быструю загрузку страниц.

Поддержка и развитие

Процесс создания сайта не ограничивается его запуском. Важный этап — его последующая поддержка и развитие. Программисты регулярно обновляют контент и функциональность, мониторят работу и анализируют данные о посещаемости и поведении пользователей.

Они реагируют на обратную связь пользователей и вносят необходимые изменения. Также разработчики продолжают развивать сайт, учитывая изменения в требованиях и технологиях.

* 1. **Создание базы данных**