

1. ОБЗОР ЛИТЕРАТУРЫ

1.1 Исследование предметной области

1.1.1 Клиент-серверная архитектура

Клиент-серверная архитектура (рисунок 1.6) – архитектура для построения веб приложений, где задачи распределяются между клиентом и сервером. При проектировании этой архитектуры важно рациональное использование вычислительных ресурсов, для оперативного отображения информации, переданной от сервера. Есть три подхода для разделения логики:

- большая часть логики реализована на сервере;
- большая часть логики реализована на клиентской стороне;
- комбинированный подход.

Взаимодействие между клиентской и серверной частью осуществляется с помощью протоколов HTTP, FTP, HTTPS, SMTP, RPC. Схема взаимодействия клиент-серверного приложения находится на рисунке 1.6.

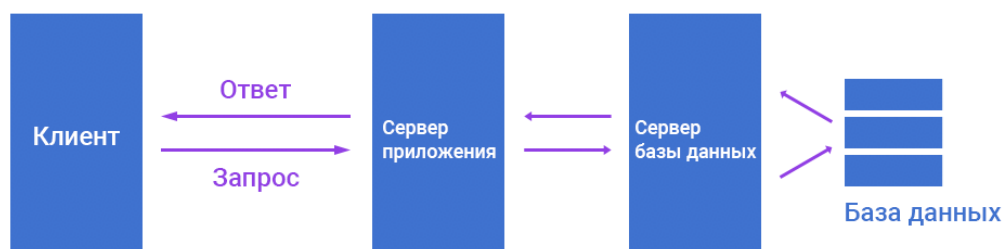


Рисунок 1.6 – Базовая схема клиент-серверных приложений

Преимущества клиент-серверной архитектуры:

- кроссплатформенность приложений;
- высокая эффективность вычислений удалённым сервером;
- лучшая защищённость данных, благодаря хранению на сервере;
- небольшая нагрузка на сеть;
- масштабируемость системы.

Недостатки клиент-серверной архитектуры:

- зависимость работоспособности всей системы от сервера;
- зависимость работоспособности системы от качества сети;
- дорогостоящее серверное оборудование;
- обслуживание оборудования требует дополнительных работников;
- большие системы требуют отдельные помещения для оборудования.

1.1.2 Архитектура веб-приложений

Веб-приложение — это приложение, состоящее из клиентской и серверной части. Клиентская часть располагается в веб-браузере и представляет собой набор веб-страниц. Серверная часть располагается на компьютере или специальном оборудовании и обслуживает запросы клиентской части.

Веб-приложение носит интерактивный характер и позволяет вводить данные, динамически обрабатывать их и получать некие результаты.

Веб-приложения разрабатываются и используются с различными целями. Они позволяют обмениваться информацией и проводить транзакции, продавать и покупать товары и услуги онлайн, совместно работать над проектами, создавать текстовые файлы, электронные таблицы и презентации, открывать доступ к ним другим пользователям.

С помощью веб-приложений можно работать с заметками, списками задач, управлять файлами в облаке, преобразовывать величины из одних единиц измерения в другие, переводить тексты, создавать парсеры и так далее – они применяются во многих различных сферах.

Функции веб-приложения:

- динамическая работа веб-страниц;
- передача и приём данных от клиента к серверу;
- доступ ко многим клиентам одновременно;
- авторизация пользователей.

Современные веб-приложения по архитектуре делятся на монолитные и микросервисные. Наглядно различия между ними изображены на рисунке 1.5.

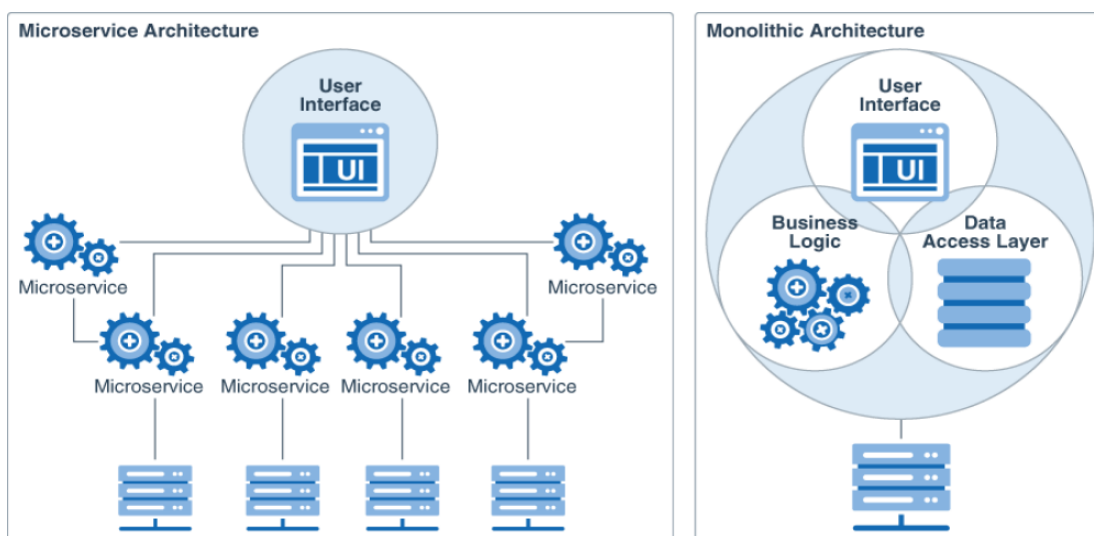


Рисунок 1.5 – Различия микросервисной и монолитной архитектуры

Монолитная архитектура – традиционный метод разработки приложений. Приложение в монолитной архитектуре разрабатывается как единое целое, идёт в одном пакете. Разработка приложения начинается с модульной многоуровневой или шестиугольной архитектуры.

Эта архитектура состоит из следующих типов слоев:

- уровень представления. Это уровень пользовательского интерфейса, который обрабатывает HTTP-запросы с использованием HTML, XML, JSON;
- уровень бизнес-логики. На этом уровне задаётся логика обработки пользовательских запросов, правила и ограничения;
- уровень доступа к базе данных. На этом уровне происходят все обращения к базе данных;
- уровень интеграции приложений. На этом уровне происходят интеграции ПО с другими системами.

Несмотря на то, что в монолитной системе имеется многоуровневая логическая архитектура, все приложения будут развёрнуты одним монолитом.

Из недостатков можно выделить:

- недостаток надлежащей модульности;
- одна кодовая база.

Микросервисная архитектура – архитектура, следующая модульному подходу для разработки разных приложений. Микросервисная архитектура включает в себя некоторое количество небольших, автономных и независимых модулей, выполняющих различные действия. Каждый сервис имеет возможность самостоятельной реализации соответствующих бизнес-единиц.

Все функциональные возможности приложения разбиты на отдельные и независимые развертываемые модули, которые взаимодействуют друг с другом с помощью методов, называемых интерфейсами прикладного программирования (API). Каждую из служб в архитектуре микросервисов можно легко масштабировать, развертывать и обновлять независимо.

Микросервисная архитектура – это слабо связанная архитектура, каждый компонент независим по отношению друг к другу. Для их написания используются различные языки программирования. Кроме того, они могут использовать другой тип хранения данных.

1.2 Обзор аналогов

При разработке приложения исследованы существующие приложения, аналогичные по функционалу, которые находятся в общем доступе в сети Интернет. Все они условно бесплатные, оплата предлагается лишь за продукты, которые планирует заказать пользователь.

1.2.1 Каталог Onliner.by

Onliner.by (рисунок 1.1) – платформа белорусского производства, предоставляющая возможности для просмотра, анализа, комментирования, повторной перепродажи различных товаров [1]. В последние годы, Onliner.by стал своеобразным «форумом» для многих белорусов.

Пользователь имеет возможность просмотреть различные товары, в том числе в одной ценовой категории, оценить различные бренды, изучить отзывы и комментарии об каком-либо продукте, зарегистрироваться и авторизоваться. Зарегистрированные пользователи получают различные бонусы, в виде скидок, бонусных баллов, кэшбэка. Доступен заказ кредитной карточки Onliner Клевер, которая предоставляет различные скидки, акционные предложения.

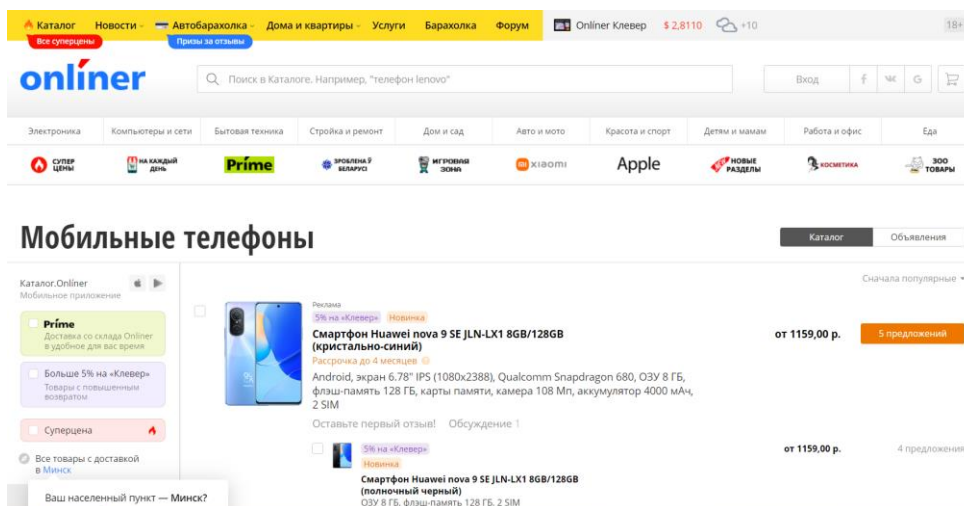


Рисунок 1.1 – Раздел мобильных телефонов сайта onliner.by

В сочетании с удобным пользовательским интерфейсом и огромным ассортиментом, пользователь может подобрать себе товар на любой вкус. Onliner.by предоставляет не только мобильную технику, но и различные бытовые товары.

Среди плюсов платформы:

- удобный пользовательский интерфейс;
- большой ассортимент товаров;
- различные скидки и бонусы;
- возможность сравнения товаров.

Основными недостатками исследуемой платформы являются:

- одноязычность;
- возможные несовпадения с ценами на странице при создании заказа.

1.2.2 A1

A1 (рисунок 1.2) – провайдер телекоммуникационных, ИКТ- и контент-услуг в Беларуси [2].

A1 – крупнейший сотовый оператор в Республике Беларусь, предоставляющий услуги не только в сфере коммуникаций, но и реализующий продукцию в виде мобильной техники.

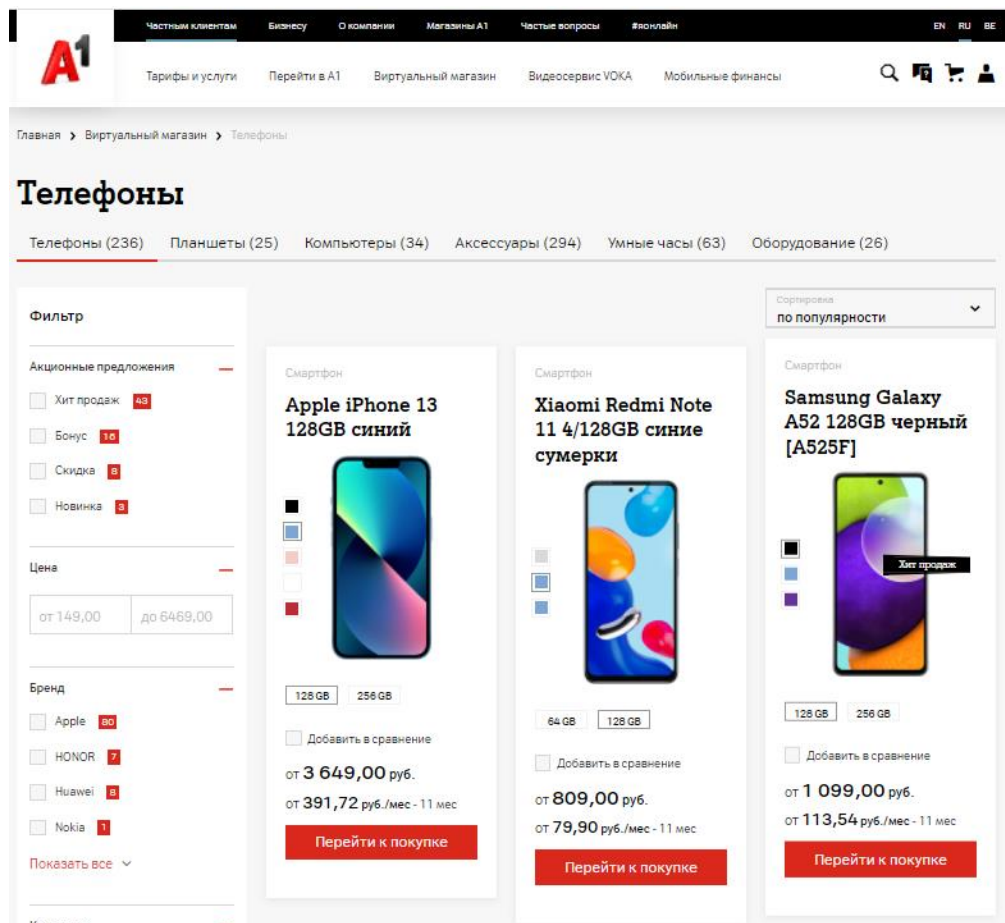


Рисунок 1.2 – Виртуальный магазин A1

Приложение имеет широкий функционал для подбора и сравнения устройств, для их визуального представления. Пользователь может выбрать один из 3 языков. Как и в приложении, представленном выше, пользователь может подобрать и сравнить варианты различной мобильной техники. Особые скидки и предложения предоставляются абонентам сотовой сети A1.

В магазине представлено множество флагманов из мира мобильной техники, а также различные аксессуары и оборудование. Присутствует возможность покупки в рассрочку, кредит.

Достоинства платформы:

- приятный и удобный пользовательский интерфейс;
- различные варианты оплаты;
- многоязычность.

Недостатки платформы:

- не представлен низкий ценовой бюджет;
- скидки только для абонентов этой сотовой сети.

1.2.3 Алло

Алло (рисунок 1.3) – белорусская сеть салонов цифровой техники [3]. Сеть входит в число лидеров по продажам мобильной техники в Республике Беларусь.

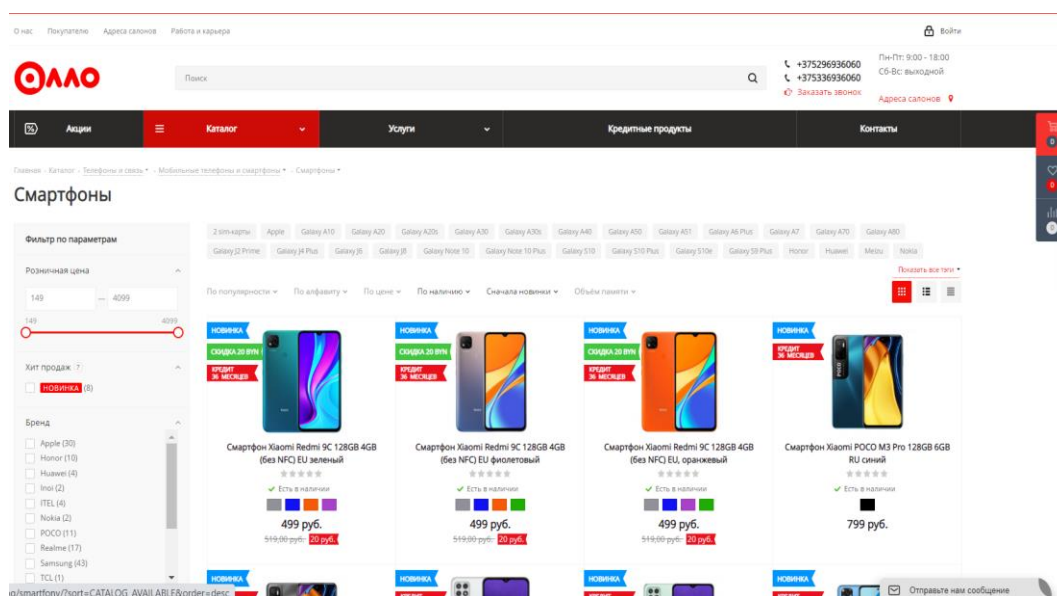


Рисунок 1.3 – Онлайн-магазин alloplus.by

Алло использует свой веб-сайт, с целью удобства для покупателей и увеличения продаж. Среди товаров: смартфоны, планшеты, гаджеты и аксессуары. Постоянные скидки и большой ассортимент продукции привлекают покупателей.

С помощью сервиса покупатель может подобрать и анализировать устройство, ознакомиться с акциями и специальными предложениями, оформить кредит и рассрочку, оформить дополнительные услуги по настройке и гарантийному обслуживанию, проверить наличие товара, а если потребуется, найти адрес ближайшего к покупателю салона (для самовывоза).

Среди преимуществ можно выделить:

- удобный пользовательский интерфейс;
- большой ассортимент продукции;

- возможность выбора дополнительных услуг;
- возможность заказа в 1 клик;
- различные варианты гарантийного обслуживания.

Присутствуют следующие недостатки:

- ошибки в работе;
- нет возможности оставлять отзывы.

1.2.4 Mobileshop

Mobileshop (рисунок 1.4) – общеевропейский портал, для покупок различных мобильных устройств [4]. Портал пользуется популярностью у жителей Евросоюза.

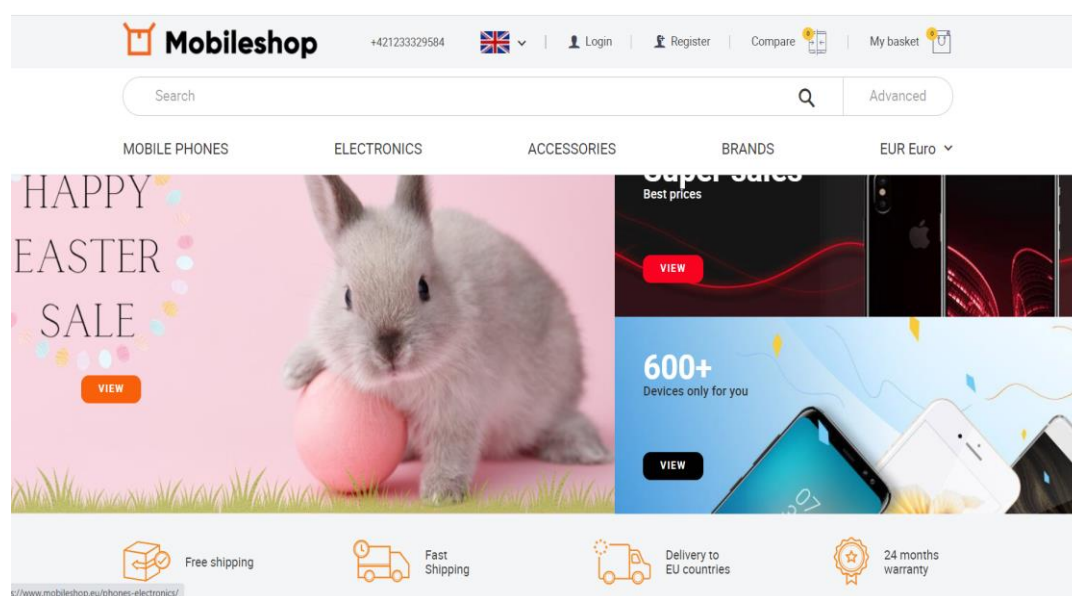


Рисунок 1.4 – Главная страница портала Mobileshop

Портал сочетает в себе функционал рассмотренных ранее аналогов, но имеет свои особенности, в виде большего количества способов оплаты, цены в различных валютах.

Преимущества:

- платформа доступна во всех странах ЕС;
- множество поддерживаемых языков;
- большее, в сравнении с предыдущими аналогами, количество способов оплаты;
- всегда бесплатная доставка.

Недостатки:

- недоступно для стран Европы, не состоящих в ЕС;
- отсутствует поддержка русского и белорусского языков.

1.2.5 Связной

Связной — российская компания, федеральная розничная сеть, специализирующаяся на продаже широкого ассортимента различных товаров, таких, как смартфоны, персональные средства связи, компьютерная техника [5].

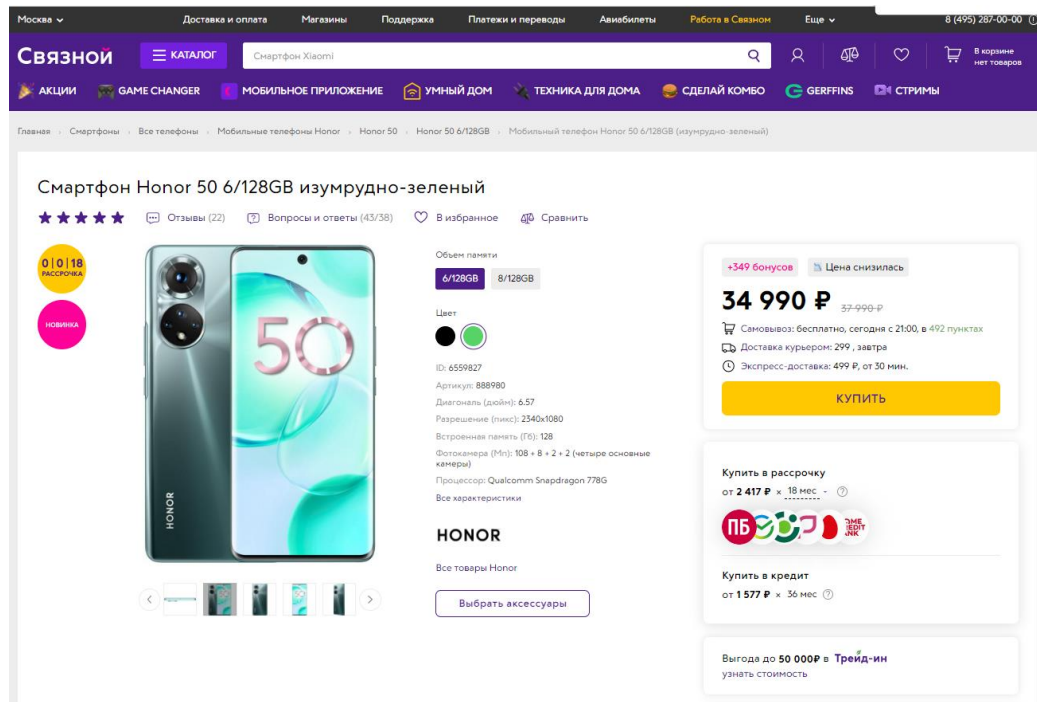


Рисунок 1.5 – Интернет-магазин мобильной техники Связной

Пользователь имеет возможность визуальнo оценить товар, сравнить его характеристики с другими товарами, выбрать подходящий ему вариант. Присутствует возможность выбрать аксессуары для будущей покупки.

Среди плюсов можно отметить:

- удобный пользовательский интерфейс;
- множество скидок и предложений;
- возможность покупки в кредит и рассрочку;
- большой ассортимент.

Минусы исследуемого аналога:

- работает только на территории РФ;
- дорогая доставка.

1.3 Обзор технологий и инструментов

1.3.1 Клиентская часть

Клиентская часть системы представляет вкладку в браузере. Существует

множество популярных технологий разработки, для выбора подходящей для проекта, следует сделать обзор доступных вариантов.

1.3.1.1 HTML

HTML (англ. HyperText Markup Language) – стандартизированный язык разметки документов для просмотра веб-страниц в браузере [6]. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

1.3.1.2 CSS

CSS (англ. Cascading Style Sheets) – формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML) [7]. CSS используется для задания шрифтов, цветов, стилей, для установления расположения различных блоков из HTML. CSS позволяет отделить описание логической структуры веб-страницы от описания её внешнего вида.

1.3.1.3 Bootstrap

Bootstrap (также известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений [8]. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. Упрощает создание шаблона и создание внешнего вида веб-страницы.

1.3.1.4 JavaScript

JavaScript – прототипно-ориентированный язык программирования [9]. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

1.3.2 Серверная часть

В клиент-серверной архитектуре, серверное программное обеспечение является «мозгом», она отвечает за получение и обработку различных запросов пользователей, например, за отправку страницы, осуществление платежей. Рассмотрим технологии используемые в разработке серверного ПО.

1.3.2.1 Java

Java – это высокоуровневый, объектно-ориентированный язык программирования, который разработали в Sun Microsystems [10].

Для языка разработаны специальные библиотеки, которые могут использоваться для решения большинства задач, связанных с взаимодействием с другими технологиями и различными протоколами.

Программы на языке Java транслируются в байт-код (определенный набор инструкций), который затем выполняется на виртуальной машине Java. Достоинством такого подхода является независимость байт-кода от операционной системы и оборудования. Байт-код может исполняться на любой вычислительной машине, для которой реализована виртуальная машина Java.

Относительная независимость виртуальной машины Java (JVM) от самого языка Java привела к разработке других языков, которые могут работать на JVM, включая Ruby (JRuby), Python (Jython), Groovy, Clojure и JavaScript.

Преимущества языка Java:

- кроссплатформенность;
- использует объектно-ориентированный подход;
- многопоточность;
- интерпретированность;
- большое количество документации;
- автоматическое управление памятью.

Недостатки:

- медленная скорость работы, в сравнении с C/C++;
- сложность кода.

1.3.2.2 Python

Python – это высокоуровневый язык программирования общего назначения, который используется в том числе и для разработки веб-приложений. Язык ориентирован на повышение производительности разработчика и читаемости кода.

Python поддерживает несколько парадигм программирования: структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. В языке реализована динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Программный код на Python организовывается в функции и классы, которые могут объединяться в модули, а они в свою очередь могут быть объединены в пакеты. Python обычно используется как интерпретируемый, но может быть скомпилирован в байт-

код Java и в MSIL (в рамках платформы .NET).

1.3.2.3 Spring Framework

Spring Framework – программный фреймворк и контейнер IoC, разработанный для платформы Java и упрощающий разработку приложений (конфигурация, тестирование) [11]. Фреймворк может использоваться для web, мобильных и настольных приложений. Spring не обеспечивает конкретную модель программирования, предоставляя свободу для разработки. Широкий набор инструментов Spring Framework позволяет повысить безопасность приложения, упростить взаимодействие с базой данных, уменьшить количество кода.

1.3.2.4 Maven

Apache Maven — фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM (англ. Project Object Model), являющемся подмножеством XML [12]. Проект Maven издаётся сообществом Apache Software Foundation.

Maven используется для построения и управления проектами, написанными на Java, C#, Ruby, Scala, и других языках.

Среди примечательных альтернатив — система автоматической сборки Gradle, построенная на принципах Apache Ant и Maven, но использующая специализированный DSL на Groovy вместо POM-конфигурации.

1.3.3 Системы управления базами данных

Одним из важнейших компонентов веб-приложения является база данных. При выборе СУБД, следует учесть такие факторы как: быстродействие, отказоустойчивость. Выбрать базу данных для разрабатываемого приложения можно из вариантов, описанных далее.

1.3.3.1 MySQL

MySQL – это реляционная система управления базами данных с открытым исходным кодом [13]. В настоящее время эта СУБД одна из наиболее популярных в веб-приложениях — подавляющее большинство использует именно MySQL (часто только её, без альтернатив), а почти все веб-фреймворки поддерживают MySQL уже на уровне базовой конфигурации (без дополнительных модулей).

Из преимуществ СУБД MySQL стоит отметить простоту использования, гибкость, низкую стоимость владения (относительно платных СУБД), а также масштабируемость и производительность.

1.3.3.2 MSSQL Server

Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основным используемым языком запросов — Transact-SQL. Из плюсов можно отметить масштабируемость, быстрое извлечение данных, автоматизация административных задач. Минусы: высокая стоимость, зависимость от операционной системы (работает на Windows).

1.3.3.3 MongoDB

MongoDB — система управления базами данных, работающая с документо-ориентированной моделью данных. В отличие от реляционных СУБД, MongoDB не требуются схемы, таблицы или отдельный язык для запросов. Информация хранится в виде коллекций или документов.

Продукт позиционируется в качестве промежуточного звена между классическими СУБД и NoSQL. MongoDB не использует схемы, как это делают реляционные базы данных, что повышает производительность всей системы.

1.4 Постановка задачи

Разрабатываемая система должна соответствовать следующим требованиям:

- стабильность работы;
- высокая производительность.

При этом стоимость разработки должна быть максимально низкой, а сроки сжатыми.

Требования, которым должно соответствовать серверное программное обеспечение:

- быстрая работа системы;
- масштабируемость;
- отказоустойчивость;
- грамотное использование вычислительных мощностей.

В соответствии с представленными выше требованиями, в проекте будет реализована возможность авторизации, оптимизированная работа с базой данных, проверки на различные ошибки и исключительные ситуации.

Клиентская часть должна быть удобной в использовании, динамичной и предоставлять следующие возможности пользователям:

- просмотр списка товаров;
- просмотр детального описания товара;
- возможность добавлять товары в корзину;
- возможность сделать заказ;
- авторизоваться.

После авторизации пользователи получают следующие функции (права администратора):

- просмотр списка заказов;
- добавление нового товара;
- редактирование существующего товара;
- управление статусом заказа.