**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«Южный федеральный университет»**

**Институт высоких технологий и пьезотехники Кафедра прикладной информатики и инноватики**

**Едленко Сергей Александрович**

**Информационный ресурс магазина строительных материалов**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

**по направлению 09.03.03 – Прикладная информатика**

**Научный руководитель – старший преподаватель**

**Яценко Дмитрий Владимирович**

**Ростов-на-Дону – 2023**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ВЫСОКИХ ТЕХНОЛОГИЙ И ПЬЕЗОТЕХНИКИ**

**Кафедра прикладной информатики и инноватики**

**З А Д А Н И Е**

**на выпускную квалификационную работу**

**Студент** гр. 4–6 Едленко С.А.

1. **Тема:** Информационный ресурс магазина строительных материалов
2. **Срок сдачи законченной работы:** –.--.2023
3. **Исходные данные:**

Государственные стандарты:

1. ГОСТ 7.1–2003. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления;
2. ГОСТ 7.82–2001 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления;
3. ГОСТ 19.001–77. Единая система программной документации (ЕСПД). Общие положения;
4. ГОСТ 19.781–90. Обеспечение систем обработки информации программное. Термины и определения;
5. ГОСТ 19.701–90 (ИСО 5807–85). Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения;
6. ГОСТ 19.103–77. Единая система программной документации. Обозначения программ и программных документов;
7. ГОСТ 19.101–77. Единая система программной документации. Виды программ и программных документов;
8. ГОСТ 19.106–78. Единая система программной документации. Требования к программным документам, выполненным печатным способом;
9. ГОСТ 19.104–78. Единая система программной документации. Основные надписи;
10. ГОСТ 19.301–79. Единая система программной документации. Программа и методика испытаний. Требования к содержанию и оформлению;
11. ГОСТ 19.402–78. Единая система программной документации. Описание программы;
12. ГОСТ 19.401–78. Единая система программной документации. Текст программы. Требования к содержанию и оформлению;
13. ГОСТ 19.504–79. Единая система программной документации. Руководство программиста. Требования к содержанию и оформлению;
14. ГОСТ 19.505–79. Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению;
15. ГОСТ 19.202–78. Единая система программной документации. Спецификация. Требования к содержанию и оформлению.

Основное требование – разработать веб-приложение, реализующее функционал интернет магазина паркета.

1. **Перечень вопросов, подлежащих разработке:**
   1. Анализ предметной области.
   2. Обоснование технических решений.
   3. Проектирование элементов системы.
   4. Реализация компонентов системы.
   5. Разработка методики тестирования.
   6. Тестирование системы.
2. **Перечень графического материала:**

Подготовка графических материалов для презентации работы.

1. **Консультанты по работе:**
2. **Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_** Яценко Д. В.

*Подпись ФИО*

1. **Задание принято к исполнению**

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

*Дата Подпись студента*

## Аннотация

к выпускной квалификационной работе на тему “Информационный ресурс магазина строительных материалов”.

Работа включает: n страниц, n таблиц, n рисунков, n приложений; использованных источников – n.

Цель работы – разработка информационного ресурса магазина строительных материалов.

В ходе работы были выбраны методология и инструменты разработки, проведен анализ существующих решений, их преимуществ и недостатков. На основе анализа были составлены требования для данного проекта и разработаны UML диаграммы классов и вариантов использования и компонентов.

В результате был создан информационный ресурс, отвечающий заявленным требованиям.

Ключевые слова: Vue.js, TypeScrypt, PHP, Laravel, PostgreSQL,

СОДЕРЖАНИЕ

1. Постановка задачи
   1. Словесная постановка задачи
   2. Формальная постановка задачи
   3. Функциональные возможности веб-приложения
   4. Анализ существующих решений
   5. Вывод по 1-й главе
2. Проектирование программного продукта
   1. Декомпозиция задачи
   2. Методология проектирования
   3. UML-моделирование
      1. Диаграмма вариантов использования
      2. Диаграмма классов
      3. Диаграмма компонентов
   4. Обоснование используемых инструментов разработки
      1. Обоснование выбора языка программирования для фронтенда
      2. Обоснование выбора языка программирования для бэкенда
      3. Обоснование выбора базы данных
      4. Обоснование выбора среды разработки
   5. Вывод по 2-й главе
3. Реализация
4. Тестирование
5. Безопасность жизнедеятельности
6. Экономическое обоснование

# **Введение**

Современному человеку тяжело представить жизнь без интернета. Он используются повсеместно, будь то образование, здравоохранение, управление бизнесом, общение или покупки. Согласно статистике The World Bank на 2020 год 60% всего населения Земли пользуется Интернетом [1], а в России это значение составляет 85% [2]. Огромная аудитория и динамичный ее рост способствуют созданию новых сервисов, услуг, интернет-магазинов, информационных ресурсов и других веб-приложений.

Нельзя не согласится, что активное развитие интернет сообщества делает нашу повседневную жизнь удобнее: государственные сервисы позволяют решить большинство вопросов, связанных с бюрократией, будь то получение справки или оплата налогов, чтобы написать кому-то сообщение или позвонить мы используем мессенджеры, а не голубиную почту, бронь билетов на любой вид транспорта доступена нам не выходя из дома, как и образование, библиотеки, онлайн-кинотеатры, магазины и многое другое. Стоит отметить, что мы этим с радостью пользуемся, так согласно опубликованной информации от Ведомости следует, что траты жителей РФ на онлайн-образование в 2021 году впервые оказались выше, чем на офлайн-сегмент и составили 226 млрд. рублей против 214 млрд. рублей [3]. Онлайн библиотеки подобным образом перенимают инициативу и на примере национальной библиотеки Республики Карелия можно убедиться, что количество онлайн (59%) посещений превышает количество физических (41%) [4]. Интернет продажи тоже увеличивают свои обороты, так по данным Росстата в 2020 году доля продаж через интернет в общем объеме оборота розничной торговли составлял 3,9%, в 2021 году он уже составляет 5,1. Тенденция роста онлайн покупок в РФ наблюдается несколько лет подряд так, в 2014 году он составлял всего 0,7%, в 2015 – 0,9%, в 2016 – 1,2% и т.д [5].

Уже не остается сомнений в тенденции перехода услуг в онлайн пространство и отсутствие возможности узнать об том или ином оффлайн сервисе через интернете или воспользоваться им вызывает, как минимум, ограниченность развития бизнеса.

Нынешние предприниматели дают себе отчет о глобальных изменениях, которые привнесло появление интернета и следуют новым тенденциям. Таким образом, наличие веб-приложения на сегодняшний день является обязательным элементом для успешного предпринимателя. Не важно какую нишу он занимает, возможность ознакомиться с предлагаемыми товарами или услугами дает четкое понимание покупателям что они могут здесь найти, тем самым, экономя свое время, время продавца, издержки предпринимателя на найм дополнительного персонала. Помимо этого, интернет, обладая воистину огромным сообществом, дает безграничные возможности продвижения, сотрудничества и реализации товаров и услуг.

Из всего вышесказанного можно сделать вывод об актуальности темы разработки информационного ресурса, что и стало предметом данной выпускной квалификационной работы.

# **ПОСТАНОВКА ЗАДАЧИ**

## Словесная постановка задачи

Результатом выпускной квалификационной работы является информационный ресурс магазина строительных материалов.

Данное веб-приложение должно предоставлять покупателям возможность удобно ознакомиться с имеющимся ассортиментом магазина и при необходимости оформить заявку на заказ. Администратор (менеджер) информационного ресурса должен обладать функционалом управления, позициями товаров (услуг) и оставленных заявок.

## Формальная постановка задачи

Учитывая предметную область и словесную постановку задачи, нужно создать информационный ресурс магазина строительных материалов.

Данный ресурс позволяет пользователю, покупателю, просмотреть каталог имеющейся продукции или услуги, при необходимости использовать фильтрацию по названию, стоимости или категории, получить основные характеристики товара в карточке товара, добавить товар (услугу) в корзину и оформить заявку на заказ для получения более подробной информации от продавца.

Продавец, он же администратор информационного ресурса имеет функционал получения заявок на заказы, администрирования наполнение каталога товаров(услуг) и сопутствующей информации о товаре (услуге), а также возможность добавления или удаления пользователей, которые смогут управлять наполнением ресурса.

Таким образом информационный ресурс состоит из двух частей. Первая - это сторона покупателя, на которой предоставлена информация об доступной продукции или возможности аренды инструмента, с возможностью оформления заявки на заказ. Вторая - сторона администратора или менеджера данного ресурса, который обеспечивает актуальность выводимой информации и обработку поступивших заказов.

Для этого потребуются следующие сущности:

* Товар - сущность, содержащая информацию о товаре
* Администратор - пользователь, который производит модерацию информационного ресурса
* Заказ - сущность, свойства которой хранят контактную информацию заказчика и информацию о выбранных товарах

## Функциональные возможности веб-приложения

## Из формальной постановки задачи можно сделать вывод, что информационный ресурс должен иметь простой интерфейс, лаконичный дизайн и реализовывать следующий функционал:

* Просмотр товаров и услуг магазина
* Фильтрация и сортировка товаров и услуг
* Поиск товаров и услуг по названию
* Просмотр личной карточки товара с подробной информацией о нем
* Добавление товаров и услуг в корзину
* Создание и отправка заявки на заказ
* Авторизация администраторов информационного ресурса
* Удаление, изменение и добавление заявок на заказ
* Удаление, изменение и добавление администраторов информационного ресурса
* Удаление, изменение и добавление товаров и услуг
* Добавление товаров и услуг из xml файла

## Анализ существующих решений

Разрабатываемый программный продукт не является уникальным и существуют решения с похожим функционалом.

Примером такого аналога может послужить облачный сервис управления торговлей и складом МойСклад [7]. В функционал данного веб-приложения входит: складской учет, обработка заказов, розничные продажи, работа с клиентами, управление финансами, подключение интернет магазина. Стоит отметить, что это полноценная система управления бизнесом и функционал у него на порядок шире, чем требует того поставленная задача.

Данный продукт имеет подписочную систему оплаты, имеется и бесплатный тариф, но он не включает в себя подключение интернет-магазина. Минимальная стоимость тарифа с возможностью подключения интернет магазина 1000 рублей + 500 рублей в месяц за подключение интернет-магазина.

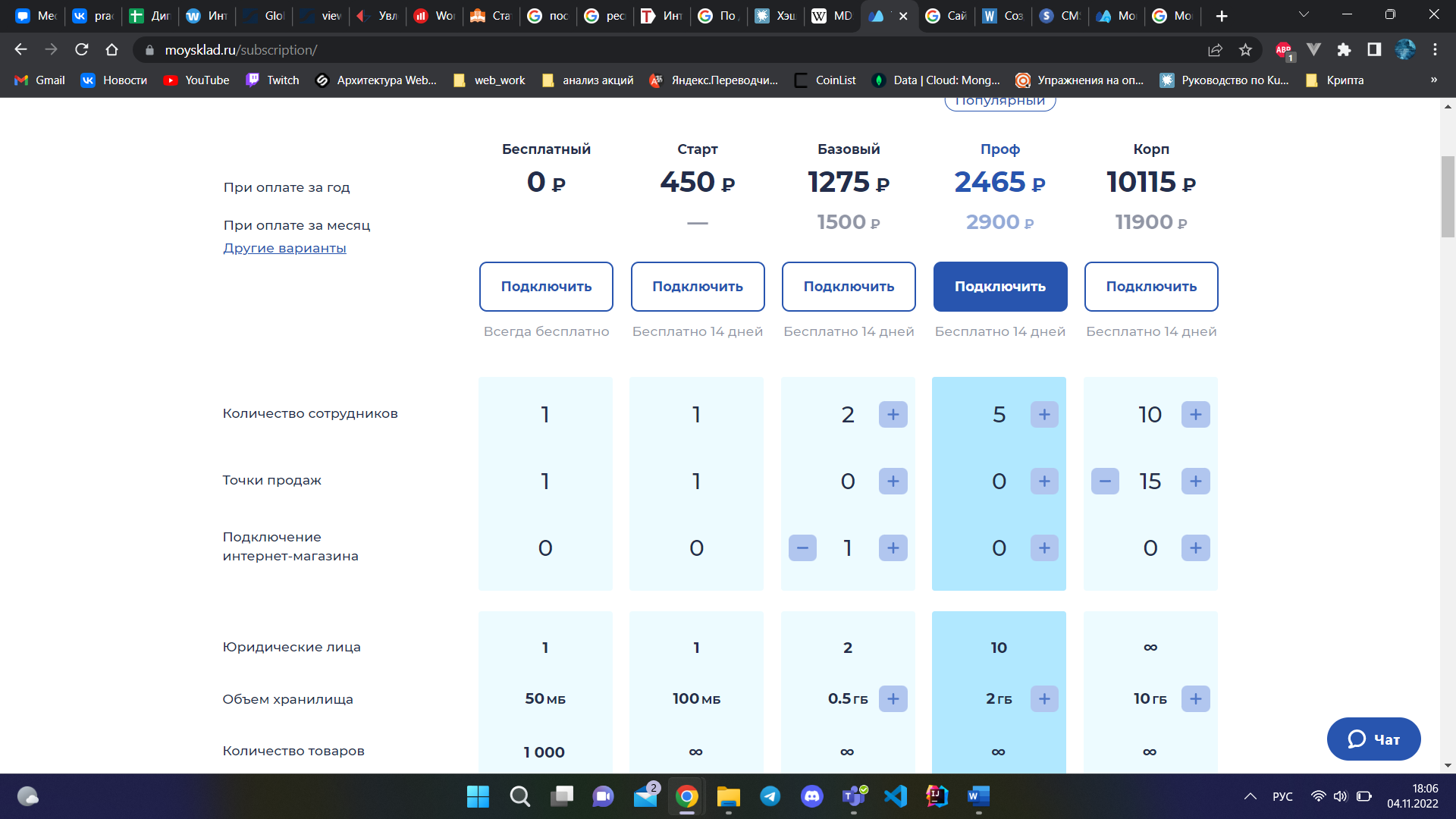


Рисунок 1 — Тарифные планы облачного сервиса управления торговлей и складом МойСклад

Еще одним примером аналога может послужить CMS система WordPress. Это своего рода конструктор сайтов, который позволяет человеку, не имеющего глубоких познаний в программировании, сконструировать свой сайт. Как можно заметить, основным преимуществом данной платформы является простота и отсутствие необходимости найма специалистов. WordPress имеет достаточное количество бесплатных тем, которыми можно свободно пользоваться, однако, бесплатные инструменты не предоставляют возможности администрирования содержимым созданного ресурса, а тем более управлять заявками пользователей. Помимо финансовой составляющей, для реализации этих задач будет необходимо вмешательство специалиста, который настроит данный функционал.

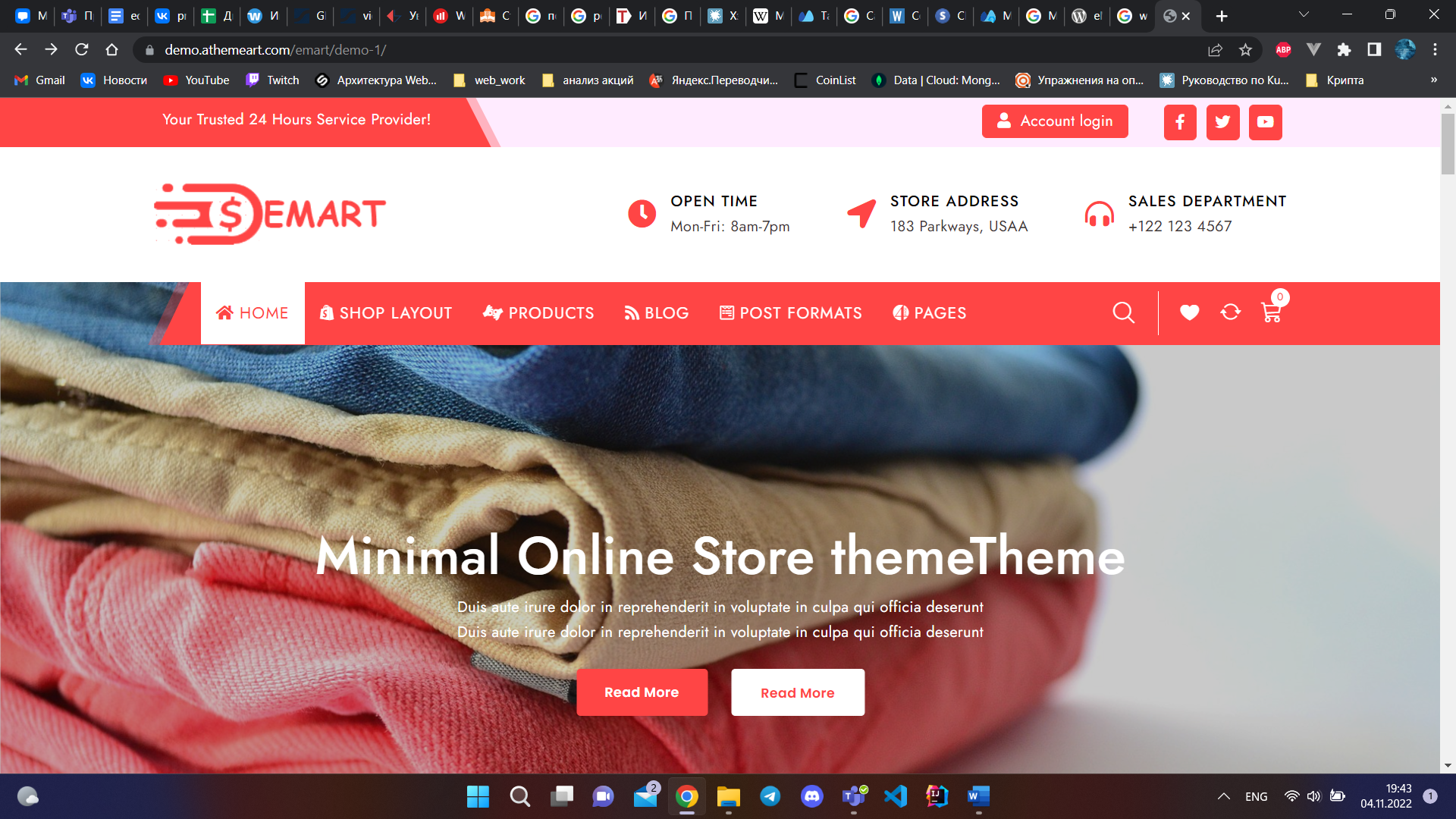


Рисунок 2 — пример интернет-магазина созданного с помощью WordPress

Таблица 1 — Сводная таблица для аналогов

| Названия  Функции | МойСклад | WordPress |
| --- | --- | --- |
| Наличие интернет-ресурса | Платная функция 500 рублей месяц | + |
| Возможность сортировки товаров | + | + |
| Названия  Функции | МойСклад | WordPress |
| Возможность администрирования содержания веб-ресурса | + | - |
| Возможность оформления заказов | + | + |
| Возможность администрирования заказов | + | - |
| Возможность создания новых администраторов веб-ресурса | + | - |
| Сложность управления системой | + | - |
| Стоимость | 1500 рублей в месяц | бесплатно |

Из вышеупомянутых сравнений можно сделать вывод, что аналоги предоставляют либо чересчур большой функционал, который нагружает пользователя и вынуждает его платить за ненужные функции, либо недостаточный, который является условно бесплатным и требует доработки специалиста.

## Вывод по 1-й главе

В первой главе были описаны словесная и формальная постановки задач, выявлены функциональные требования к информационной системе и приведён сравнительный анализ аналогичных систем, представленных на рынке.

# **ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА**

## Декомпозиция задачи

Поставленную задачу можно разбить на ряд подзадач:

1. Выбор методологии проектирования
2. Разработка UML диаграмм
3. Выбор инструментов разработки
4. Выбор дизайна пользовательского интерфейса

### Методология проектирования

Для разработки информационного ресурса был использован объектно-ориентированная методология анализа и проектирования [8].

В основе объектно ориентированный подхода проектирования лежит объектная декомпозиция, в которой статическая структура системы описывается объектами, которые имеют собственное поведение, имитирующее модель реального мира, и взаимосвязями между ними, а поведение системы – в способах обмена сообщения между объектами.

Таким образом объектно ориентированный подход проектирования включает 3 ключевых понятия:

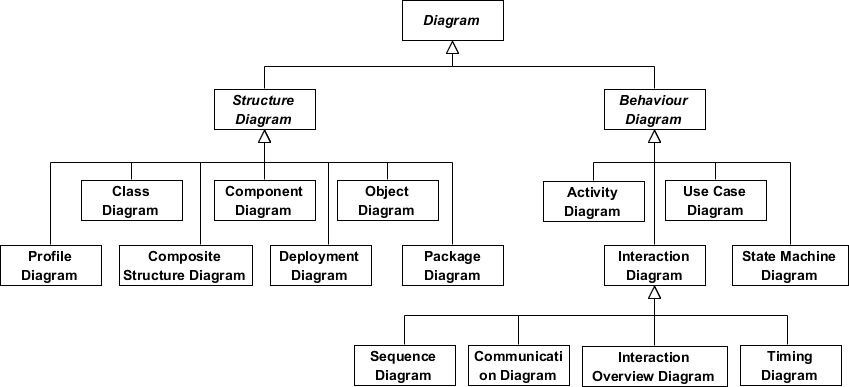
* ООА (object oriented analysis) - объектно-ориентированный анализ, представляющий собой создание модели функциональных требований системы с помощью объектно-ориентированного подхода. Это значит, что в ходе данного подхода формируются требования вокруг объектов, которые объединяют как поведение, так и состояния, смоделированные на основе объектов реального мира [9].
* OOD (object oriented design) - объектно-ориентированное проектирование. Является методологией проектирования, основанный на объектной декомпозиции и представления логической и физической, а также статической и динамической моделей проектируемой системы [10].
* OOP (object oriented programming) - объектно-ориентированное программирование. Это парадигма программирования, основанная на представлении программы в виде взаимодействующих объектов [11].

Основными принципами объектно-ориентированного анализа и проектирования являются:

* Инкапсуляция. Принцип защиты некоторый компонентов объекта от внешних сущностей.
* Наследование. Возможность класса расширять или переопределять функциональность другого класса.
* Абстракция. Выделение значимых характеристик объекта, которые отличают его от всех других объектов.
* Полиморфизм. Свойство методов с одним названием выполнять различные функции в зависимости от принадлежности к тому или иному классу [11].

## UML - моделирование

UML (Unified Modeling Language) – стандартизированный язык моделирования, состоящий из интегрированного набора диаграмм, предназначенный предоставить стандартную нотацию, которую можно использовать во всех объектно-ориентированных системах [12].



# Рисунок 1 – Структурная схема существующих видов UML диаграмм

UML диаграммы подразделяются на структурные диаграммы и диаграммы поведения.

Структурные диаграммы описывают статические структуры системы и ее части на разных уровнях абстракции, их реализацию и взаимоотношения между собой.

К структурным диаграммам относятся:

* Диаграмма классов
* Диаграмма компонентов
* Диаграмма объектов
* Диаграмма композитной/составной структуры
* Диаграмма пакетов
* Диаграмма развёртывания
* Диаграмма профилей

Диаграммы поведения описывают динамику объектов в системе, которые можно описать как серию изменений в системе с течением времени [12].

Диаграммами поведения считаются:

* Диаграмма вариантов использования
* Диаграмма взаимодействия
* Диаграмма последовательности
* Диаграмма сотрудничества
* Диаграмма состояний
* Диаграмма деятельности
* Диаграмма автомата
* Диаграмма синхронизации

Инструментом для моделирования UML-диаграмм стал PlantUML. Это инструмент с открытым исходным кодом, который позволяет удобно в текстовой форме моделировать UML-диаграммы. Помимо этого, для некоторых сред разработки, например Eclipse или Intellij Idea существуют плагины, которые позволяют создавать диаграммы непосредственно из модели существующего проекта [14].

### Диаграмма вариантов использования

Диаграмма вариантов использования (use cases diagram) описывает модель с точки зрения функциональности системы и ее окружения. Данная диаграмма позволяет связать требования системы и то, каким образом она удовлетворяет данные требования.

Основными компонентами диаграммы вариантов использования являются:

* Случаи применения
* Актеры
* Отношения между актерами и прецедентами или между прецедентами и прецедентами. К базовым отношениям относятся:
  + Ассоциация – отражает отношение между актером и прецедентами.
  + Включение – описывает включение поведения одного прецедента в качестве составной части в поведении другого прецедента.
  + Расширение – описывает возможность одного варианта использования быть дополненными свойствами другого расширенного варианта использования.
  + Обобщение – описывает, что некоторый прецедент может быть обобщен до другого прецедента, то есть являться потомком.

Далее, на рисунке 2 и в таблице 2, приведена диаграмма вариантов использования и детализация диаграммы вариантов использования информационного ресурса магазина строительных материалов.

# 

# Рисунок 2 – Диаграмма вариантов использования информационного ресурса магазина строительных материалов

Таблица 2 – Детализация диаграммы вариантов использования

| Вариант использования | Детализация |
| --- | --- |
| authorization(авторизация) | Авторизация администратора для дальнейшей работы с инструментами администрирования веб-ресурса. Осуществляется посредством формы, при успешной авторизации пользователь перенаправляется на главную страницу администрирования ресурсом. В противном случае, пользователь оповещается о некорректно введенных данных  **Входные параметры**: логин, хеш пароля  **Выходные параметры**: успешно / неуспешно |
| manage catalog of goods (управление каталогом товаров) | Администрирование каталога товаров. Подразумевает создание, удаление, редактирование позиций каталога товаров. Также можно обновить данные каталога, загрузив xml файл с актуальными данными.  **Входные параметры**: id товара, действие, данные о товаре  **Выходные параметры**: статус запроса успешно / неуспешно |
| manage list of orders (управление списком заказов) | Администрирование списка заказов. Оставленные заказчиками заказы могут быть удалены или изменены администратором. Также может быть создан и описан новый заказ администратором.  **Входные параметры**: id заказа, действие, данные о заказе  **Выходные параметры**: статус запроса успешно / неуспешно |
| view catalog of goods | Просмотр каталога товаров. Ресурс предоставляет удобно ознакомиться с предоставляемыми товарами и услугами, отфильтровать по цене и отсортировать по категориям позиции, посмотреть подробную информацию о товаре и добавить в корзину.  **Входные параметры**: список товаров  **Выходные параметры**: нет |
| create order (создать заказ) | Создание заявки на заказ. Пользователь вводит свои контактные данные, номер телефона и email, после чего отправляет свою заявку на заказ с выбранными позициями продавцу.  **Входные параметры**: список выбранных товаров, email заказчика, номер телефона заказчика  **Выходные параметры**: статус запроса успешно / неуспешно |

### 

### Диаграмма классов

Диаграмма классов (class diagram) – это структурная диаграмма, описывающая общую структуру иерархии классов, отношения между ними, свойства, методы и их кооперацию [13].

Основными взаимоотношениями в диаграмме классов являются:

* Ассоциация – это отношение описывает, что один класс системы может быть связан с другим классом.
* Наследование – указывает, что один класс является частным случаем другого.
* Агрегация – это вариативность ассоциации, описывает, что один класс может быть контейнером или списком других классов. Причем, время существования хранимых классов не взаимосвязано с временем существования класса-контейнера.

Ниже, на рисунке 3, приведена диаграмма классов информационного ресурса магазина строительных материалов.

# 

Рисунок 3 – Диаграмма классов информационного ресурса магазина строительных материалов

### 

### Диаграмма компонентов

Диаграмма компонентов (component diagram) – диаграмма, позволяющая описать особенности физического представление системы, определить архитектуру разрабатываемой системы и установить зависимости между программными компонентами и исполняемым кодом.

Основными элементами диаграммы компонентов являются:

* Компонент – это элемент модели, представляющий модульную часть системы с инкапсулированным содержимым.
* Предоставляемый интерфейс – интерфейс, необходимый компоненту для выполнения заявленного функционала.
* Соединители – элемент, описывающий взаимосвязь между двумя и более компонентами.

Далее, на рисунке 4, изображена диаграмма компонентов разрабатываемого программного продукта.

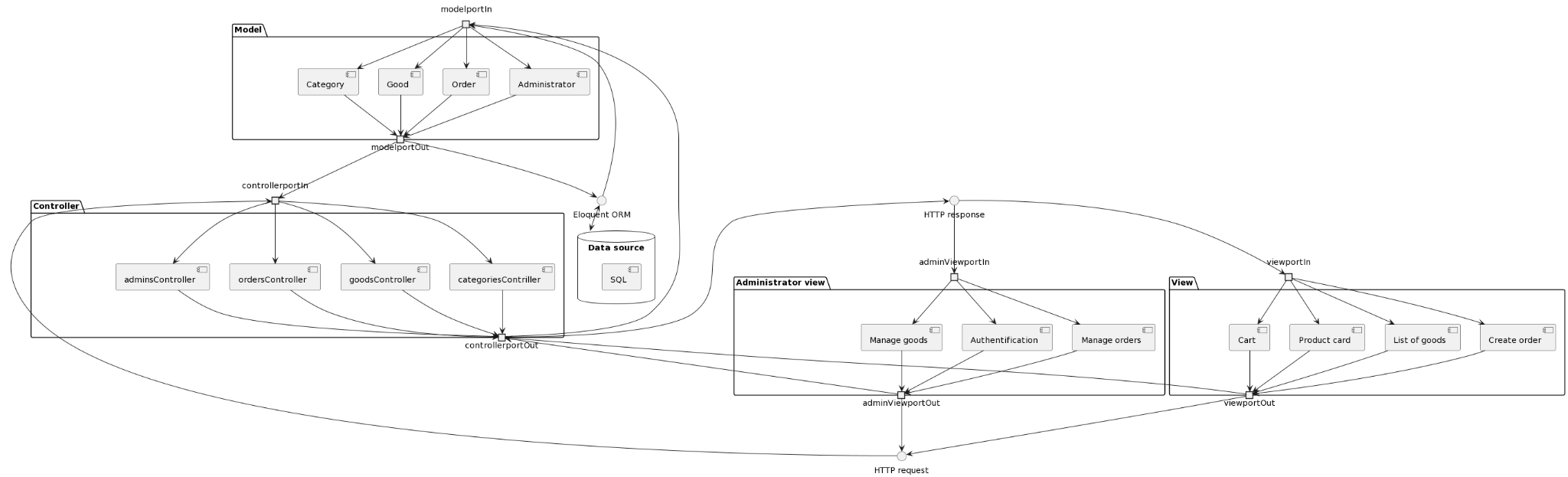


Рисунок 4 – Диаграмма компонентов информационного ресурса магазина строительных материалов

Программный продукт реализует MVC (Model-View-Controller) паттерн, основной задачей которого является отделить бизнес-логику от интерфейса. Благодаря этому приложение лучше масштабируется, тестируется, поддерживается и реализуется.

Основными составляющими MVC модели являются:

* View (представление) – обеспечивает отображение данных пользователю, полученных из контроллера и модели.
* Model (модель) – бизнес-логика приложения и включает методы выборки ORM.
* Controller (контроллер) – связующее звено модели и представления. Отвечает за обработку запросов пользователя.

## Обоснование используемых инструментов разработки

Выбор инструментов разработки является важным этапом в проектировании приложения. От выбора зависят скорость разработки, т.к. незнание, например, языка программирования добавляют время на изучение особенностей данного языка, следовательно увеличивают время и качество разработки. Также стоит подметить, что каждый инструмент имеет свои достоинства в тех или иных обстоятельствах и может показывать лучшие параметры производительности для поставленной задачи нежели другой язык программирования в тех же условиях.

Далее будет проанализирован рынок предоставляемых решений и будет сделан выбор инструментов разработки в соответствии с поставленной задачей и личными предпочтениями и навыками разработчика.

### Обоснование выбора языка программирования для фронтенда

Фронтэнд (frontend) – это видимая часть приложения, с которой пользователь может взаимодействовать: кнопки, страница сайта, ссылки, чекбоксы и другого рода элементы управления.

На 2022 год, согласно статистике, подавляющее часть рынка фронтэнд языков программирования занимают javaScript и TypeScript [15].

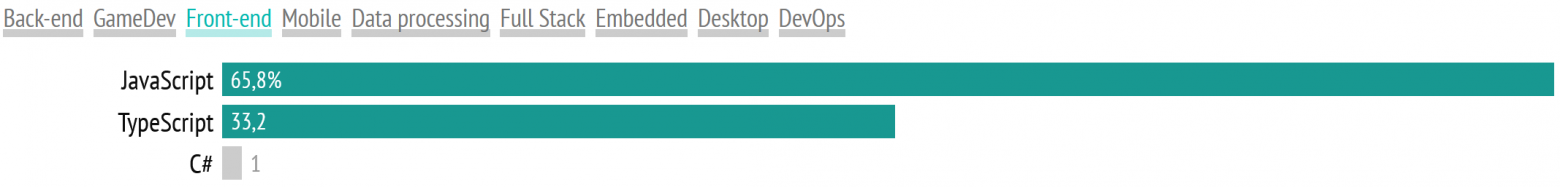


Рисунок 5 – Статистика популярности языков программирования в фронтенд разработке на 2022 год

javaScript — легковесный, компилируемый, объектно-ориентированный язык программирования, который помогает реализовать сложное поведение web-страниц, создавать адаптивные и интерактивные элементы на странице.

TypeScript – язык программирования для web приложений, основанный на javaScript и расширяющий его возможности.

TypeScript является надмножеством javaScript и имеет тот же функционал и компилируется в простой JavaScript. Кроме того TypeScript имеет ряд различий:

* TypeScript – объектно-ориентированный язык программирования. Может использовать классы, интерфейсы, наследование.
* TypeScript – работает не только на стороне клиента, но и на стороне сервера.
* TypeScript – имеет статическую проверку типа.

Главным преимуществом TypeScript является статическая поддержка типов, что позволяет контролировать типы данных и четко понимать что нужно положить и что мы получим. Это позволяет избежать множество ошибок, следовательно сократить время на их исправление, и сделать приложение более надежным, т.к. вероятность возникновения ошибок в ходе эксплуатации уменьшается.

Данное преимущество стало решающим в выборе языка программирования для фронтэнд части приложения.

Помимо языка программирования стоит обратить внимание на выбор фреймворка или библиотеки, которая позволит сделать разработку быстрее и удобнее.

На данный момент распространены 3 инструмента, совместимых c TypeScript:

* React – это библиотека с открытым исходным кодом, которая позволяет синхронизировать компоненты интерфейса с данными приложения, однако для полноценной разработки знание одного React недостаточно: еще необходимо знать Babel, Webpack, Redux и другие инструменты в зависимости от проекта [16].
* Angular – открытая платформа для разработки веб приложений нацеленный на разработку одностраничных решений, написанная на языке TypeScript. Данный фреймворк имеет строгую архитектуру, рассчитанную на сложные проекты, и хорошо масштабируется, однако, это является как достоинством, так и недостатком. Это усложняет разработку маленьких или средних проектов. Помимо этого к достоинствам можно отнести двунаправленное связывание, это значит, что данные могут быть связаны с отображаемыми элементами и изменение данных или отображаемого элемента приведет к изменению второго.
* Vue — javaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Так же как и React использует виртуальную модель DOM, компоненты и двустороннее связывание, но главной отличием является архитектура. Она разработана таким образом, что позволяет разработчикам переносить уже существующие проекты во фреймворк, подключать дополнительный функционал по очереди, а не все сразу. Как уже стало ясно Vue.js и React имеют много общего за счет использования виртуальной модели DOM, но и различий у них не мало, а именно методы для рендеринга в DOM.Vue используются HTML-шаблоны и JSX, а React — только JSX.

Из приведенного сравнения можно сделать вывод, что освоение React недостаточно для полноценной разработки и помимо самой библиотеки необходимо знание как минимум Babel, Webpack, Redux, что увеличит время изучения, следовательно и разработки.

Angular также имеет ряд особенностей, связанных со структурой разрабатываемого продукта, что является его преимуществом, но для информационного ресурса магазина строительных материалов, который мы разрабатываем, это недостаток, так как этот продукт не подразумевает объемности и сложности проекта, в которых данный фреймворк хорош.

Vue является оптимальным решением, так как этот фреймворк имеет все необходимые инструменты для полноценной разработки, преимущества Angular, относительно двустороннего связывания, качественной документации и помимо этого имеет самый низкий порог входа, относительно представленных вариантов.

Таким образом, TypeScript и Vue были выбраны как инструменты разработки для фронтэнд части проекта.

### Обоснование выбора языка программирования для backend

Бэкенд (backend) – это бизнес-логика приложения, он отвечает за взаимодействие пользователя с внутренними данными приложения, которые отображаются во фронтенде.

По аналогии с выбором фронтенд языка программирования, обратимся к статистике.

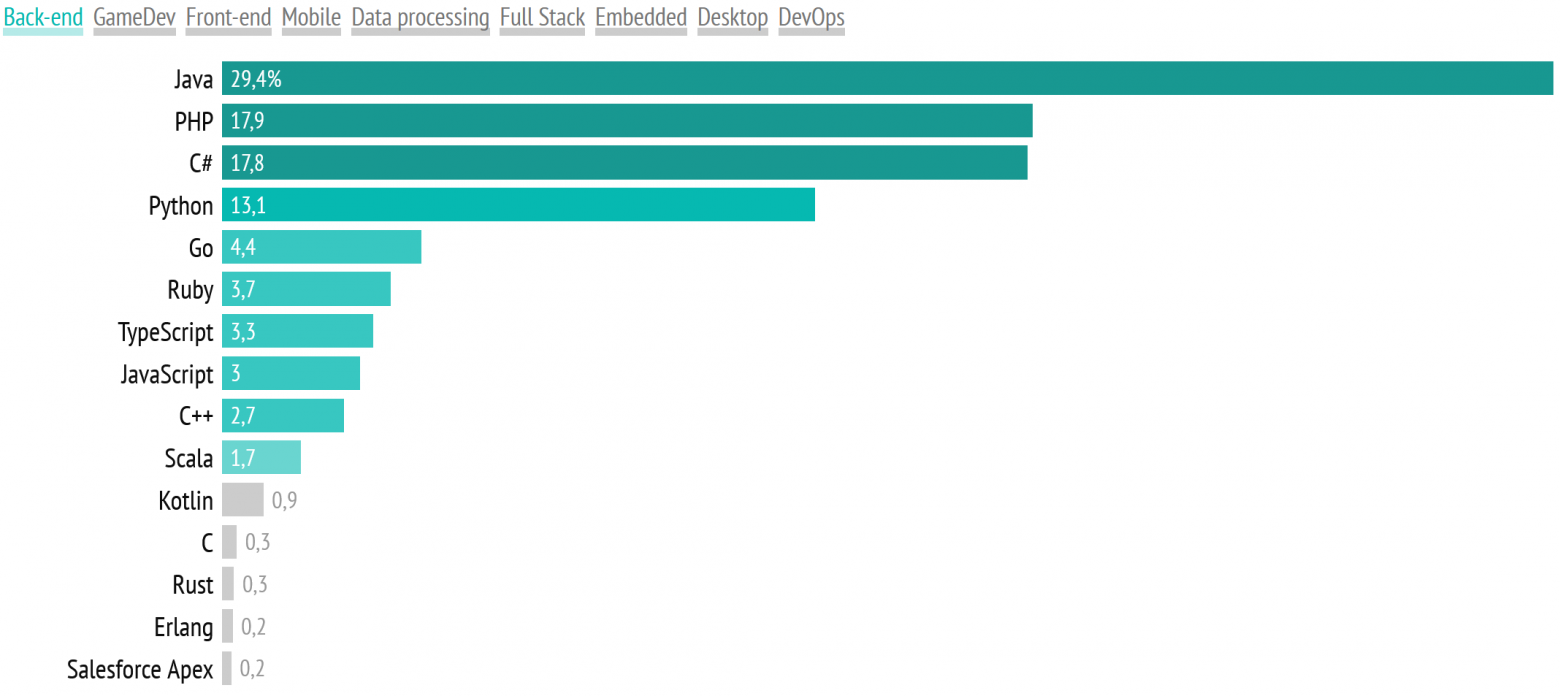


Рисунок 5 – Статистика популярности языков программирования в бэкенд разработке на 2022 год

На рисунке 5 можно заметить, что наиболее популярными языками программирования для бэкенд являются Java, PHP, C# и Python.

Основными критериями выбора языка программирования для бэкенда будут: возможность быстро начать разработку, наличие хорошей документации и большого сообщества.

* Java – объектно-ориентированный язык со строгой типизацией, являющийся универсальной программной платформой. Этот язык используют для разработки настольных, сетевых и мобильных приложений, а так же в web-разработке. К достоинствам данного языка можно отнести кроссплатформенность, надежность, объектно-ориентированный подход. Недостатками являются: низкая скорость и производительность, сложный синтаксис.
* PHP – один из самых распространённых языков программирования в веб-разработке, его используют около 77,6% всех сайтов [17]. К достоинствам данного языка программирования можно отнести простоту установки и настройки, большое сообщество, так как язык образовался в далеком 1995 году, PHP допускает ошибки, код выполняется пока не достигнет участка с ошибкой. К недостаткам относится динамическая типизация.
* Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью. Достоинствами языка являются: интерпретируемость и гибкость языка, большое сообщество, многофункциональность. Недостатки: низкая производительность, динамическая типизация.

Из приведенного описания можно сделать вывод, что Java не подходит, так как имеет достаточно сложный синтаксис, что замедлит обучение. PHP и Python имеют достаточно низкий порог входа, большое сообщество, но одним из минусов Python является низкая производительность, помимо этого PHP используется в 77,6% веб-приложений [17], что является показателем предрасположенности языка к веб-разработке и большим количеством решений тех или иных проблем, возникающих при разработке.

Помимо самого PHP будет использован фреймворк laravel 9, который предназначен для разработки с использованием архитектурной модели MVC, которая задействована в нашем приложении. Данный фреймворк имеет ряд готовых решений, которые ускорят разработку и ее качество:

* Механизм регистрации и аутентификации
* Валидоторы
* Eloquent ORM
* Механизм очередей
* Интерфейс для Cron задач
* Удобный механизм роутинга
* Работа с сессиями

Таким образом для бэкенд части разрабатываемого приложения будут использованы PHP и фреймворк laravel 9.

### Обоснование выбора базы данных

Реляционная база данных – это набор таблиц, между которыми установлены определенные взаимосвязи. Для обслуживания реляционной базы данных и создания запросов к ней система управления базой данных использует язык структурированных запросов (Structured Query Language, SQL) [18].

Для нормального функционирования приложения необходимо хранилище данных. В качестве такого хранилища выступит реляционная база данных. На рынке представлено множество различных продуктов, я сделаю выбор из наиболее популярных из них, так как это означает наличие того или иного обучающе-справочного материала по ним.

Выбор будет осуществлен из следующих баз данных:

* OracleDB – самое известное и надежное решение на сегодняшний день. Занимает первое место в рейтинге DB-Enginers [19]. Преимуществами данной РСУБД являются: поддержка со стороны компании Oracle, хорошая масштабируемость, безопасность, эффективность в управлении памятью. Однако недостатки, также присутствуют, к ним относятся дороговизна лицензии, порядка 17 500 долл. США. за процессор для стандартной версии и сложная документация.
* MySQL – РСУБД с открытым исходным кодом. На данный момент находится под управлением Oracle. К достоинствам можно отнести большую пользовательскую базу, отличную поддержку, основной функционал данного решения является бесплатным, однако за дополнительную плату можно получить рад дополнительных функций, быстрая настройка и быстрый старт. К недостаткам можно отнести плохую масштабируемость при использовании бесплатного функционала, долгое создание инкрементных резервных копий или изменение архитектуры данных в MySQL.
* PostgreSQL – объектно-реляционная база данных. Это означает, что все данные в ней представлены в виде объектов, а не столбцов и строк. Главными достоинствами данного решения являются отличная масштабируемость, безопасность, полностью бесплатный функционал, отличная документация. К недостаткам можно отнести то, что поддержку данного решения предоставляют вендорские компании, сложность системы и ее настройки, в простых операциях чтения PostgreSQL уступает в производительности конкурентам.

Проанализировав возможные варианты, был сделан выбор в пользу MySQL, так как это решение обладает достаточным бесплатным функционалом для закрытия потребностей разрабатываемого приложения, имеет большую пользовательскую базу, что означает наличие документации, обучающих ресурсов и обсуждений проблем, также на выбор повлияла возможность быстрого начала разработки и относительно быстрой настройки.

### Обоснование выбора среды разработки

Интегрированная среда разработки (Integrated development environment — IDE) – комплекс программных средств, используемых для разработки программного обеспечения.

Необходимо, чтобы в IDE была поддержка Typescript, Vue, PHP и Laravel 9. Данные требования удовлетворяют NetBeans и PHPStorm.

NetBeans является бесплатным продуктом, имеется поддержка русского языка, NetBeans поддерживает все известные фреймворки, в том числе Zend, Laravel, Doctrine, CakePHP, Smarty, Yii и Symfony2. Кроме того, поддерживаются javascript, Typescript, HTML, CSS и PHP в самых свежих версиях и сопутствующие фреймворки Vue, React, Angular.

PHPStorm – платная программная среда, но для студентов и преподавателей профильных ВУЗов поставляется бесплатно, для проектов с открытым исходным кодом. Отлично подходит для full-stack веб-разработки. Также поддерживает фреймворки Symfony, Zend, Yii, CakePHP, основные CMS системы Magento, Drupal, Wordpress, отладчики Zend Debugger и Xdebug, а для фронтенда доступны CSS3, HTML5, Sass, CoffeeScript, TypeScript, Stylus, Less, React, Angular, Vue.

Так как имеется опыт работы, возможность использовать PHPStorm и он удовлетворяет необходимым требованиям, выбор остановим на нем.

## Вывод по 2-й главе

Во второй главе была произведена декомпозиция задачи, выбрана методология проектирования, смоделированы UML-диаграммы вариантов использования, классов и компонентов, и обоснован выбор используемых инструментов разработки. Благодаря этим шагам можно приступить к реализации программного продукта.

# Реализация

## -

## -

## -

## -

## -

## -

## -

# Источники

1. Статистика использования интернета физическими лицами в мире. URL: <https://data.worldbank.org/indicator/IT.NET.USER.ZS> (Дата обращения 27.10.2022)
2. Статистика использования интернета физическими лицами в России. URL: <https://data.worldbank.org/indicator/IT.NET.USER.ZS?locations=RU> (Дата обращения 27.10.2022)
3. Статистика розничных продаж электронной коммерции в мире. URL: <https://www.insiderintelligence.com/content/worldwide-ecommerce-growth-drops-single-digits-while-overall-retail-muddles-through> (Дата обращения 27.10.2022)
4. Траты жителей РФ на онлайн образование превысили офлайн сегмент. URL: <https://www.vedomosti.ru/media/articles/2022/06/01/924747-rossiyane-onlain-obrazovanie> (Дата обращения 29.10.2022)
5. Статистика посещений национальной библиотеки Республики Карелии за 3 квартал 2022 года. URL: <http://library.karelia.ru/Statistika> (Дата обращения 29.10.2022)
6. Доля продаж через Интернет в общем объеме оборота розничной торговли РФ. URL:<https://rosstat.gov.ru/storage/mediabank/Internet_torgovlya.xls> (Дата обращения 29.10.2022)
7. Облачный сервис управления торговлей и складом МойСклад. URL: <https://www.moysklad.ru/vozmozhnosti/> (Дата обращения 04.11.2022)
8. Grady Booch, Robert Maksimchuk, Michael Engle, Jim Conallen, Kelli Houston, Bobbi Young Ph.D. Object-Oriented Analysis and Design with Applications, Pearson Education, 2007. 721с.
9. Объектно-ориентированный анализ и дизайн. URL: <https://ru.wikibrief.org/wiki/Object-oriented_analysis_and_design> (Дата обращения 07.12.2022)
10. Объектная модель. URL: <https://www.helloworld.ru/texts/comp/other/oop/ch02.htm> (Дата обращения 07.12.2022)
11. Объектно-ориентированное программирование. URL: <https://en.wikipedia.org/wiki/Object-oriented_programming> (Дата обращения 07.12.2022)
12. What is Unified Modeling Language (UML)? URL: [https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/#use-case-diagram) (Дата обращения 09.12.2022)
13. Class diagram URL: <https://en.wikipedia.org/wiki/Class_diagram> (Дата обращения 10.12.2022)
14. Plugins URL: <https://plantuml.com/ru/sitemap-plugins> (Дата обращения 10.12.2022)
15. Рейтинг языков программирования 2022. URL: <https://habr.com/ru/post/651585/> (Дата обращения 15.12.2022)
16. React, Vue или Angular. URL: <https://ru.hexlet.io/blog/posts/react-vue-ili-angular> (Дата обращения 15.12.2022)
17. Usage statistics of PHP for websites. URL: <https://w3techs.com/technologies/details/pl-php> (Дата обращения 17.12.2022)
18. Ключевые аспекты при выборе базы данных для вашего приложения. URL: <https://habr.com/ru/company/otus/blog/562852/> (Дата обращения 17.12.2022)
19. DB-Engines Ranking. URL: <https://db-engines.com/en/ranking> (Дата обращения 17.12.2022)

UI макет <https://www.figma.com/file/c4lUU4IScsHbLWGkcIfKFF/Furniture-Shop---Freebie-(Community)?node-id=0%3A1>