Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Волгоградский государственный технический университет»

|  |  |
| --- | --- |
| Факультет | Электроники и вычислительной техники |
| Кафедра | Программное обеспечение автоматизированных систем |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Согласовано | | | | | | | | |  | Утверждаю | | | | | | | | |
|  | | | | | | | | |  | и. о. зав. кафедрой | | | | | | | | |
| (должность гл. специалиста предприятия) | | | | | | | | |  |
|  | | | |  |  | | | |  |  | | | |  | О. А. Сычев | | | |
| (подпись) | | | |  | (инициалы, фамилия) | | | |  | (подпись) | | | |  | (инициалы, фамилия) | | | |
| « |  | » |  | | | 20 |  | г. |  | « |  | » |  | | | 20 |  | г. |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| к | выпускной квалификационной работе бакалавра | | | | | | | | | | | | | | | на тему |
| (наименование вида работы) | | | | | | | | | | | | | | | | |
| Разработка веб-сервиса для просмотра и сравнения рейтингов и отзывов | | | | | | | | | | | | | | | | |
| автосервисов по городам России | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | |  | | | | | |  |  | | | | | | | |
| Автор | |  | | | | | |  | Гайлевич Ян Валерьевич | | | | | | | |
|  | | (подпись и дата подписания) | | | | | |  | (фамилия, имя, отчество) | | | | | | | |
| Обозначение | | | ВКРБ–09.03.04–10.19–05–24 | | | | | | |  | | | | | | |
|  | | | (код документа) | | | | |  | | | | | | | | |
| Группа | | | ПрИн-466 | | | | |  | | | | | | | | |
|  | | | (шифр группы) | | | | |  | | | | | | | | |
| Направление | | | 09.03.04 – Программная инженерия,  Разработка программно-информационных систем | | | | | | | | | | | | | |
|  | | | (код и наименование направления, наименование программы (профиля)) | | | | | | | | | | | | | |
| Руководитель работы | | | | |  | | | | | | | |  | | Гилка В.В. | |
|  | | | | | (подпись и дата подписания) | | | | | | | |  | | (инициалы и фамилия) | |
| Консультанты по разделам: | | | | | | | | | | | | | | | | |
|  | | | | | |  |  | | | | |  | |  | | |
| (краткое наименование раздела) | | | | | |  | (подпись и дата подписания) | | | | |  | | (инициалы и фамилия) | | |
|  | | | | | |  |  | | | | |  | |  | | |
| (краткое наименование раздела) | | | | | |  | (подпись и дата подписания) | | | | |  | | (инициалы и фамилия) | | |
| Нормоконтролер: | | | |  | | | | | | |  | Кузнецова А.С. | | | | |
|  | | | | (подпись и дата подписания) | | | | | | |  | (инициалы и фамилия) | | | | |

Волгоград 2024 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Волгоградский государственный технический университет»

|  |  |
| --- | --- |
| Кафедра | Программное обеспечение автоматизированных систем |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Утверждаю | | | | | и. о. зав. кафедрой | | | |
|  |  | | | |  | О. А. Сычев | | | |
| (подпись) | | | |  | (инициалы, фамилия) | | | |
|  | « |  | » |  | | | 20 |  | г. |

**Задание**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| на | выпускную квалификационную работу бакалавра | | | | | | | | | | | | | | |
|  | (наименование вида работы) | | | | | | | | | | | | | | |
| Студент | | | Гайлевич Ян Валерьевич | | | | | | | | | | | | |
|  | (фамилия, имя, отчество) | | | | | | | | | | | | | | |
| Код кафедры | | | | 10.19 | Группа | | | | ПрИн-466 | | | |  | | |
| Тема | | Разработка веб-сервиса для просмотра и сравнения рейтингов и отзывов | | | | | | | | | | | | | |
| автосервисов по городам России | | | | | | | | | | | | | | | |
| Утверждена приказом по университету | | | | | | « | 01 | » | | сентября | 20 | 23 | | г. № | 1074-ст |
| Срок представления готовой работы (проекта) | | | | | | | | | |  | | | | | |
|  | | | | | | | | | | (дата, подпись студента) | | | | | |
| Исходные данные для выполнения работы (проекта) | | | | | | | | | | | | | | | |
| Задание, выданное научным руководителем кафедры «ПОАС» | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| Содержание основной части пояснительной записки | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| Перечень графического материала | |
| 1) |  |
|  | |
| 2) |  |
|  | |
| 3) |  |
|  | |
| 4) |  |
|  | |
| 5) |  |
|  | |
| 6) |  |
|  | |
| 7) |  |
|  | |
| 8) |  |
|  | |
| 9) |  |
|  | |
| 10) |  |
|  | |
| 11) |  |
|  | |
| 12) |  |
|  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель работы (проекта) | |  | |  | | Гилка В.В. | |
|  | | (подпись и дата подписания) | |  | | (инициалы и фамилия) | |
| Консультанты по разделам: | |  | | | |  | |
|  |  | |  | |  | |  |
| (краткое наименование раздела) |  | | (подпись и дата подписания) | |  | | (инициалы и фамилия) |
|  |  | |  | |  | |  |

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Кафедра «Программное обеспечение автоматизированных систем»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ:  и. о. зав. кафедрой ПОАС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.А. Сычев  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 г. |

Разработка веб-сервиса для просмотра и сравнения рейтингов и отзывов автосервисов по городам России

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ВКРБ–09.03.04–10.19–05–24–81

Листов 72

|  |  |
| --- | --- |
|  | Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гилка В.В.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_20 г. |
| |  | | --- | | Нормоконтролер  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузнецова А.С.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г. | | |  | | --- | | Исполнитель  студент группы ПрИн-467  \_\_\_\_\_\_ Гайлевич Ян Валерьевич «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 20 г. | |

Волгоград, 2024 г.

Аннотация

Настоящий документ является пояснительной запиской к выпускной квалификационной работе бакалавра на тему: «Разработка веб-сервиса для просмотра и сравнения рейтингов и отзывов автосервисов по городам России».

В работе обосновывается актуальность выбранной темы, проводится анализ существующих аналогов и программных решений, представлены способы применения программы, а также описан процесс проектирования, разработки и тестирования веб-сервиса.

Документ включает в себя страниц - 67, рисунков - 16, приложений -3

Ключевые слова: веб-сервис, разработка веб-сервиса, база данных, сервер, frontend, backend, Node JS, Express JS, React, автосервис, рейтинг, отзыв, СУБД.

Содержание

[Введение 8](#_Toc168329327)

[1 Анализ проблемы и существующих решений 10](#_Toc168329328)

[1.1 Характеристика предметной области 10](#_Toc168329329)

[1.2 Необходимые теоретические сведения 12](#_Toc168329330)

[1.3 Описание существующих способов решения задачи 14](#_Toc168329331)

[1.4 Перспективные подходы к решению задачи 17](#_Toc168329332)

[1.5 Сравнение с аналогами 19](#_Toc168329333)

[1.5.1 Критерии анализа 19](#_Toc168329334)

[1.5.2 Описание аналогов 20](#_Toc168329335)

[1.5.3 Таблица результатов анализа аналогов 25](#_Toc168329336)

[2 Предлагаемый способ решения поставленной задачи 27](#_Toc168329337)

[2.1 Анализ инструментов разработки 27](#_Toc168329338)

[2.2 Обоснования выбора инструментов разработки 31](#_Toc168329339)

[Выводы 34](#_Toc168329340)

[3 Архитектура, функциональная реализация и тестирование мобильного приложения 35](#_Toc168329341)

[3.1 Архитектура веб-сервиса 35](#_Toc168329342)

[3.2 Требования к функциональным характеристикам 35](#_Toc168329343)

[3.3 Требования к нефункциональным характеристикам 37](#_Toc168329344)

[3.4 Варианты использования разрабатываемого веб-сервиса 38](#_Toc168329345)

[3.5 Проектирование веб-сервиса 40](#_Toc168329346)

[3.6 Проектирование базы данных 43](#_Toc168329347)

[3.7 Проектирование классов 44](#_Toc168329348)

[3.8 Ручное тестирование веб-сервиса 45](#_Toc168329349)

[Выводы 49](#_Toc168329350)

[4 Оценка и результаты эффективности мобильного приложения 50](#_Toc168329351)

[4.1 Методы оценки эффективности мобильного приложения 50](#_Toc168329352)

[4.2 Опросное анкетирование пользователей по эффективности веб-сервиса 53](#_Toc168329353)

[Выводы 61](#_Toc168329354)

[Заключение 62](#_Toc168329355)

[Список использованных источников 63](#_Toc168329356)

[Приложение А](#_Toc168329357) - [Справка о результатах проверки выпускной квалификационной работы на наличие заимствований 65](#_Toc168329358)

[Приложение Б](#_Toc168329359) - [Техническое задание 66](#_Toc168329360)

[Приложение В](#_Toc168329361) - [Руководство системного программиста 67](#_Toc168329362)

# Введение

В современном мире, где технологии занимают центральное место в повседневной жизни, важность и актуальность надежной и удобной цифровой инфраструктуры неоспорима. Особенно это касается области автомобильных услуг, где потребность в качественном обслуживании и доступной информации о таких услугах неуклонно растет. В условиях ускоренной урбанизации и наращивания автопарка, потребители все чаще ищут эффективные и надежные способы для поддержки и обслуживания своих транспортных средств. Таким образом, веб-сервисы, предоставляющие информацию о рейтингах и отзывах автосервисов, становятся неотъемлемой частью рынка автомобильных услуг.

Важность таких веб-сервисов обусловлена не только растущим спросом на качественное автообслуживание, но и повышенной конкуренцией среди автосервисов. Потребители стремятся получить лучшие услуги по оптимальным ценам, а наличие доступной и достоверной информации о различных сервисах позволяет им сделать обоснованный выбор. В то же время, для автосервисов наличие положительных отзывов и высоких рейтингов является мощным инструментом привлечения новых клиентов и укрепления своего положения на рынке.

С учетом этих факторов, разработка веб-сервиса для просмотра и сравнения рейтингов и отзывов автосервисов по городам России представляется актуальной и перспективной задачей. Такой сервис должен не только обеспечивать доступ к надежной информации, но и предлагать удобный и интуитивно понятный интерфейс, адаптированный для различных устройств. Кроме того, важным аспектом является обеспечение конфиденциальности пользовательских данных и высокого уровня защиты от несанкционированного доступа.

Целью работы является обеспечение пользователей всей необходимой и актуальной информацией для принятия обоснованных решений при выборе автосервиса в различных городах России.

Задачи:

* произвести анализ предметной области;
* произвести обзор существующих аналогов и выявить их основные преимущества и недостатки;
* определить требования к разрабатываемому веб-сервису;
* исходя из требований произвести проектирование базы данных;
* протестировать разработанный веб-сервис и доказать его работоспособности и эффективность.

Объектом исследования в работе является процесс предоставления и получения информации о качестве услуг автосервисов в различных городах России.

Предметом исследования является разработка веб-сервиса, который обеспечивает сбор, систематизацию, визуализацию и доступ к данным о рейтингах и отзывах автосервисов.

Методы исследований. Для решения поставленных задач были использованы методы математического моделирования, системного анализа, программной инженерии, объектно-ориентированного программирования, технологии проектирования человеко-машинного взаимодействия.

Практическая ценность работы заключается в том, что веб-сервис предоставит пользователям удобный и надежный инструмент для получения объективной информации о качестве и надежности автосервисов. Это позволит потребителям принимать более обоснованные решения при выборе сервиса, основываясь на реальных отзывах и оценках других клиентов. Стимулирует предприятия к повышению качества обслуживания и уровня сервиса, так как положительные отзывы и высокие рейтинги напрямую влияют на привлекательность сервиса для потенциальных клиентов.

# 1 Анализ проблемы и существующих решений

# 1.1 Характеристика предметной области

Сравнение и анализ отзывов и рейтингов автосервисов по городам России — это сложная и многогранная тема, которая охватывает различные аспекты работы автосервисов и их взаимодействия с клиентами. Этот процесс включает сбор, обработку и интерпретацию данных о качестве услуг и уровне удовлетворённости клиентов, что требует систематического и тщательного подхода.

Рейтинги автосервисов формируются на основе агрегированных данных из разных источников, таких как специализированные сайты, социальные сети, форумы и мобильные приложения, и учитывают различные критерии, например, качество ремонта, скорость обслуживания, вежливость персонала, прозрачность ценообразования и наличие дополнительных услуг. Эти критерии позволяют получить объективную и полную картину качества предоставляемых услуг и помочь клиентам сделать осознанный выбор.

Анализ отзывов клиентов играет ключевую роль в понимании реального опыта обращения в автомобильные сервисы и выявлении типичных проблем и недостатков. Отзывы предоставляют ценную информацию о взаимодействии клиентов с персоналом, уровне профессионализма мастеров, условиях ожидания и результатах выполненных работ. Важно определить достоверность и репрезентативность отзывов, так как в интернете могут встречаться поддельные или заказные отзывы, искажающие реальность. Для этого используются методы фильтрации и анализа текстов, направленные на выявление аномалий и шаблонных сообщений, характерных для недобросовестных отзывов. Методы машинного обучения и обработки естественного языка могут значительно повысить точность и эффективность анализа, позволяя автоматизировать процесс и выявлять скрытые тенденции и паттерны в больших объёмах данных.

При сравнении автосервисов в разных городах России необходимо учитывать региональные особенности, такие как экономическая ситуация, уровень конкуренции, наличие специализированных компаний по прокату автомобилей и предпочтения местных жителей. В крупных городах, таких как Москва и Санкт-Петербург, конкуренция между автосервисами выше, что может приводить к более высоким стандартам обслуживания и, соответственно, к более положительным отзывам. В то же время, в небольших городах и сельских районах выбор автосервисов может быть ограничен, что влияет на требования и ожидания клиентов. Региональные различия также могут касаться ценовой политики и доступности определённых услуг, что важно учитывать при сравнении.

Кроме того, необходимо учитывать временной аспект, так как качество услуг и уровень удовлетворённости клиентов могут меняться со временем. Периодический мониторинг и анализ отзывов позволяют выявлять тенденции и динамику изменений, что способствует более точной оценке и сравнению автосервисов. Это особенно важно в контексте введения новых технологий и методов ремонта, изменения ценовой политики и улучшения сервисных условий. Систематическое обновление данных и их анализ в динамике позволяют выявлять не только текущее состояние рынка, но и прогнозировать будущие изменения и адаптироваться к ним.

Таким образом, сравнение и анализ отзывов и рейтингов автосервисов по городам России требуют комплексного подхода к сбору, интерпретации и оценке данных. Необходимо учитывать множество факторов, влияющих на мнения и оценки клиентов, а также специфику рынка автосервисов в разных регионах. Глубокий анализ позволяет не только выявлять лучшие практики и лидеров рынка, но и помогать автосервисам улучшать качество своих услуг, ориентируясь на реальные потребности и ожидания клиентов. Это способствует повышению общей удовлетворённости клиентов и укреплению доверия к автосервисам, что является ключевым фактором их долгосрочного успеха. В результате, повышение качества обслуживания и более точное соответствие ожиданиям клиентов могут привести к увеличению клиентской базы, улучшению репутации и, в конечном итоге, к устойчивому развитию автосервисного бизнеса.

# 1.2 Необходимые теоретические сведения

Веб-сервис – это программный компонент или набор протоколов и стандартов, которые позволяют различным программным приложениям взаимодействовать друг с другом через Интернет. Веб-сервисы предоставляют средства для обмена данными между различными системами, написанными на различных языках программирования и работающими на различных платформах.

Автосервис — это предприятие, предоставляющее разнообразные услуги по техническому обслуживанию и ремонту автомобилей. Услуги включают диагностику, устранение неполадок, замену и установку запчастей, регулярное техобслуживание, кузовные работы, покраску, шиномонтаж и балансировку колёс, а также дополнительные работы, такие как установка аксессуаров и доработка автомобилей. Автосервисы могут быть специализированными, работающими только с определёнными марками автомобилей, или мультибрендовыми, обслуживающими разные модели и марки. Ключевыми факторами эффективности работы автосервиса являются качество предоставляемых услуг, квалификация персонала и использование современного оборудования.

Рейтинг автосервиса — это интегральный показатель, отражающий общее качество услуг, основанный на мнениях и отзывах клиентов. Он может быть представлен в виде числового значения (например, от 1 до 5 звёзд) или индекса, учитывающего множество параметров. В основе рейтинга лежат оценки по разным критериям, таким как качество ремонта, скорость обслуживания, вежливость и профессионализм персонала, прозрачность ценообразования и наличие дополнительных услуг. Рейтинги формируются на основе данных, собранных с разных платформ, включая онлайн-отзывы, опросы клиентов и экспертные оценки.

Отзыв клиента — это письменное мнение о полученных услугах в автосервисе, выраженное на специализированных платформах, социальных сетях, форумах или через опросы. Отзывы содержат описание опыта взаимодействия с персоналом, качества выполненных работ, условий ожидания, стоимости услуг и общей удовлетворённости. Отзывы могут быть положительными или отрицательными и часто содержат конкретные примеры и детали, помогающие другим потребителям сформировать представление о сервисе. Важно учитывать достоверность отзывов и их репрезентативность для обеспечения объективности и точности информации.

Критерии оценки — это параметры, по которым клиенты оценивают качество услуг автосервиса. Основные критерии включают качество ремонта, скорость обслуживания, доброжелательность персонала, прозрачность ценообразования и наличие дополнительных услуг. Эти критерии помогают клиентам сформировать обоснованное мнение и принять осознанное решение при выборе автосервиса.

Агрегированные данные — это собранные и обобщённые сведения из разных источников, используемые для формирования рейтингов автосервисов и анализа отзывов клиентов. Источники включают специализированные сайты для отзывов, социальные сети, форумы, мобильные приложения и результаты опросов. Агрегация данных позволяет получить более полное и объективное представление о качестве услуг автосервисов, поскольку учитывает мнения и отзывы с разных платформ и каналов.

Фальшивые отзывы — это недостоверные или заказные отзывы, созданные с целью улучшить или ухудшить репутацию автосервиса. Такие отзывы не отражают реальный опыт клиентов и могут ввести потребителей в заблуждение. Выявление и фильтрация фальшивых отзывов важны для обеспечения достоверности рейтингов и отзывов.

Frontend, или клиентская часть, — это то, что видит и с чем взаимодействует пользователь на экране своего устройства. Это интерфейс веб-сервиса, который отображается в браузере. Основная цель фронтенда - обеспечить удобный и приятный интерфейс для пользователя, позволяя ему взаимодействовать с функционалом веб-сервиса.

Backend, или серверная часть, — это то, что происходит "за кулисами" веб-сервиса. Пользователи не видят и не взаимодействуют напрямую с бэкендом, но он играет ключевую роль в обработке данных и управлении логикой приложения. Основная задача бэкенда - обеспечить работу веб-сервиса, обрабатывать запросы от фронтенда и взаимодействовать с базой данных.

# 1.3 Описание существующих способов решения задачи их критические места, достоинства и недостатки

Рассмотрение существующих способов решения задачи, помогает понять, какие технологии и подходы будут наиболее эффективны для разрабатываемого проекта.

1. Классический подход с использованием серверных языков программирования.

Этот подход включает использование традиционных серверных языков программирования для создания серверной части веб-приложения. Включает создание API для взаимодействия с базой данных и клиентской частью.

Достоинства:

- проверенная временем технология;

- широкая поддержка и документация;

- хорошая производительность.

Недостатки:

- могут возникать сложности с асинхронной обработкой запросов;

- более высокая сложность в поддержке масштабируемости;

- трудности при работе с реальными временем обновлениями данных.

2. Использование серверного фреймворка для обработки запросов и взаимодействия с базой данных.

Этот метод предполагает использование специализированного фреймворка, который упрощает создание серверной части, управление базой данных и обработку запросов.

Достоинства:

- упрощает разработку за счет встроенных инструментов и библиотек;

- повышает скорость разработки;

- часто включает средства для работы с асинхронными запросами.

Недостатки:

- может быть сложным для новичков;

- ограниченная гибкость из-за зависимостей от фреймворка;

- возможны проблемы с производительностью при высокой нагрузке.

3. Монолитная архитектура.

В монолитной архитектуре все компоненты веб-сервиса объединены в одно большое приложение. Это включает серверную часть, клиентскую часть и базу данных.

Достоинства:

- простота в разработке и развертывании;

- легкость в тестировании и отладке;

- единый код базы, что упрощает управление.

Недостатки:

- плохая масштабируемость;

- сложности при добавлении новых функций и поддержке старых;

- в случае ошибки может страдать весь сервис.

4. Микросервисная архитектура.

В этой архитектуре веб-сервис делится на несколько независимых микросервисов, каждый из которых отвечает за свою часть функциональности.

Достоинства:

- отличная масштабируемость;

- легкость в обновлении отдельных частей системы без влияния на остальные;

- повышенная надежность и отказоустойчивость.

Недостатки:

- высокая сложность разработки и деплоя;

- необходимость настройки взаимодействия между микросервисами;

- потенциальные проблемы с производительностью из-за межсервисной коммуникации.

Критические места:

- производительность: некоторые подходы могут не справляться с высокой нагрузкой, что приводит к задержкам и снижению качества обслуживания;

- масштабируемость: важно, чтобы сервис мог расти вместе с увеличением числа пользователей и данных. Монолитные системы часто имеют проблемы с масштабируемостью;

- поддержка и обновление: легкость внесения изменений и добавления новых функций сильно зависит от выбранной архитектуры. Монолиты сложнее обновлять, чем микросервисы;

- сложность разработки: некоторые подходы требуют больше времени и усилий на начальной стадии разработки, особенно если используются сложные фреймворки или микросервисы;

- безопасность: необходимо обеспечить высокий уровень безопасности, особенно при работе с пользовательскими данными и отзывами. Каждый подход имеет свои методы обеспечения безопасности, но они могут быть трудными в реализации.

После рассмотрения существующих подходов к решению задачи можно сказать, классические методы могут быть проще в реализации, но менее гибкими и масштабируемыми.

# 1.4 Перспективные подходы к решению задачи

Также рассмотрения требуют перспективные подходы к решению задачи.

Один из подходов заключается в использовании современной архитектуры для серверной части. Этот подход предполагает создание гибкой и масштабируемой системы, которая способна обрабатывать множество запросов одновременно. Основное преимущество здесь — высокая производительность и возможность горизонтального масштабирования. Однако критическим местом может стать сложность настройки и управления системой, особенно для начинающих разработчиков.

Другой перспективный подход — это применение продвинутой системы управления базами данных. Это позволяет эффективно хранить и обрабатывать большие объемы данных, обеспечивая быстрый доступ к отзывам и рейтингам. Достоинством этого подхода является высокая скорость обработки запросов и надежность хранения данных. Однако недостатком может стать необходимость оптимизации запросов и структуры данных для предотвращения снижения производительности при увеличении объема данных.

Для клиентской части можно использовать современные библиотеки для создания пользовательского интерфейса. Это обеспечивает высокую скорость разработки и возможность создания интерактивных и отзывчивых интерфейсов. Преимущество этого подхода — улучшенный пользовательский опыт и кроссплатформенная совместимость. Критическим местом здесь может быть сложность изучения и использования библиотек для новичков, а также возможные проблемы с производительностью на старых устройствах.

Использование централизованного управления состоянием приложения также является перспективным подходом. Это позволяет легко управлять состоянием всего приложения и обеспечивает согласованность данных между различными компонентами. Основное достоинство — упрощение разработки и поддержки кода. Однако недостатком может стать усложнение архитектуры приложения, особенно если проект небольшой и не требует сложного управления состоянием.

Еще один подход — это интеграция системы для работы с асинхронными запросами. Это позволяет быстро и эффективно обрабатывать данные, получаемые с сервера, и отображать их на клиентской части. Достоинство — повышение производительности и улучшение пользовательского опыта за счет быстрой загрузки данных. Критическое место — необходимость дополнительного внимания к обработке ошибок и управлению асинхронными операциями, что может усложнить разработку.

Использование современных и перспективных подходов в разработке веб-сервиса поможет значительно повысить его качество и функциональность. Эти методы включают в себя применение новейших технологий и инструментов, следование передовым практикам в области веб-разработки, а также внедрение инновационных решений для улучшения пользовательского опыта. Благодаря этим подходам, создаваемый веб-сервис будет выгодно выделяться на фоне существующих аналогов, предлагая пользователям уникальные возможности, более высокую производительность и надежность. Такой подход также способствует более эффективному и гибкому процессу разработки, что позволяет быстрее адаптироваться к изменяющимся требованиям рынка и предпочтениям пользователей.

# 1.5 Сравнение с аналогами

# 1.5.1 Критерии анализа

В процессе анализа веб-сервисов для просмотра рейтингов и отзывов автосервисов в городах России, важно принимать во внимание ряд ключевых критериев. Вот некоторые из них:

1. Удобство использования — этот фактор определяет, насколько просто и интуитивно понятно пользователю взаимодействовать с веб-сервисом. Он включает в себя удобство навигации по сайту или приложению, понятный интерфейс, лёгкость поиска и сравнения автомобильных услуг, а также наличие функций для выполнения необходимых задач, таких как отображение информации об услугах, оценка или выбор параметров сортировки.

2. Доступность и актуальность данных — этот фактор определяет, насколько быстро и регулярно обновляется информация на веб-сервисе. Важно, чтобы данные о рейтингах и отзывах об автомобильных услугах были актуальными и соответствовали текущей ситуации на рынке. Также важно, чтобы веб-сервисом можно было пользоваться в любое время и из любого места.

3. Возможности фильтрации и сортировки — этот фактор определяет наличие и простоту использования инструментов для фильтрации и сортировки данных об автосервисах. Пользователи могут искать по разным критериям, таким как расстояние до места, стоимость услуг, рейтинги, специализация и другие параметры, чтобы быстро находить подходящие варианты.

4. Возможность оставлять отзывы и ставить рейтинги — этот фактор определяет наличие возможности оставлять отзывы и оценивать работу автосервисов. Пользователи должны иметь возможность делиться своим мнением о качестве услуг, что помогает другим пользователям принимать обоснованные решения при выборе сервиса.

5. Достоверность отзывов и рейтингов — этот фактор определяет надёжность и полезность представленных отзывов и оценок. Важно, чтобы отзывы были объективными и информативными и предоставляли реальную дополнительную ценность для потенциальных пользователей. Также важно, чтобы существовали механизмы для выявления и удаления недостоверных или вводящих в заблуждение отзывов.

6. Возможность сравнения автосервисов — этот фактор определяет способность пользователей сравнивать отзывы о разных автосервисах, чтобы принять более обоснованное решение. Пользователи должны иметь возможность сравнивать несколько сервисов одновременно, чтобы быстро увидеть различия и выбрать наиболее подходящий вариант.

7. Геолокационные функции — этот фактор определяет наличие функций для определения местоположения пользователя и поиска автосервисов поблизости. Геолокация позволяет пользователям быстро находить ближайшие сервисы, что повышает удобство использования веб-сервиса, особенно в ситуациях, когда требуется срочный ремонт или техническое обслуживание.

Эти критерии позволят определить, насколько подходит веб-сервис для поиска автосервисов в городах России, учитывая индивидуальные предпочтения и нужды пользователей.

# 1.5.2 Описание аналогов

Яндекс.Карты:

1. Удобство использования: Яндекс.Карты предоставляют простой и интуитивно понятный интерфейс для пользователей. Они легко доступны как на десктопе, так и на мобильных устройствах, что делает их удобными для использования в любой ситуации.

2. Доступность и актуальность данных: Картографические данные Яндекса обычно актуальны и широко доступны. Однако в отдаленных районах могут быть проблемы с точностью и актуальностью данных.

3. Возможности фильтрации и сортировки: Пользователи могут фильтровать объекты на карте по различным категориям, таким как автосервисы, рестораны, магазины и другие. Однако возможности сортировки ограничены, что может затруднить поиск в больших городах.

4. Возможность оставлять отзывы и ставить рейтинги: В Яндекс.Картах пользователи могут оставлять отзывы и ставить рейтинги объектам на карте. Однако механизм модерации отзывов не всегда эффективен, что может привести к появлению недостоверной информации.

5. Качество отзывов и рейтингов: сервис не надежен в обеспечении качества отзывов и рейтингов.

6. Сравнение рейтингов: В Яндекс.Картах нет возможности сравнить рейтинги нескольких автосервисов одновременно, что затрудняет выбор между различными вариантами.

7. Геолокационные функции: Геолокационные функции в Яндекс.Картах позволяют пользователям быстро находить ближайшие автосервисы, что удобно при срочном ремонте или обслуживании. Однако в некоторых случаях может быть недостаточно точной определение местоположения.

2ГИС:

1. Удобство использования: 2ГИС предоставляет удобный и простой в использовании интерфейс, который позволяет пользователям легко находить информацию о различных объектах в городе.

2. Доступность и актуальность данных: Данные в 2ГИС обычно актуальны и доступны для пользователей. Однако в некоторых регионах качество данных может быть ниже, чем в других.

3. Возможности фильтрации и сортировки: 2ГИС предоставляет широкие возможности для фильтрации и сортировки объектов по различным критериям, что упрощает поиск автосервисов.

4. Возможность оставлять отзывы и ставить рейтинги: В 2ГИС пользователи могут оставлять отзывы и ставить рейтинги объектам на карте. Однако механизм модерации отзывов может быть не всегда эффективным.

5. Качество отзывов и рейтингов: сервис не обеспечивает качество отзывов и рейтингов под вопросом.

6. Сравнение рейтингов: В 2ГИС нет возможности сравнивать рейтинги нескольких автосервисов одновременно, что может затруднить выбор между различными вариантами.

7. Геолокационные функции: Геолокационные функции в 2ГИС позволяют пользователям быстро находить ближайшие автосервисы, что удобно в ситуациях, когда требуется срочный ремонт или обслуживание. Однако точность определения местоположения может быть недостаточной.

Uremont:

1. Удобство использования: Uremont предоставляет простой и понятный интерфейс, который позволяет пользователям быстро находить информацию о различных автосервисах.

2. Доступность и актуальность данных: Данные в Uremont обновляются регулярно и обычно актуальны. Однако в некоторых регионах может быть ограниченный выбор автосервисов.

3. Возможности фильтрации и сортировки: Uremont предоставляет возможности для фильтрации и сортировки автосервисов по различным параметрам, что облегчает поиск подходящего варианта.

4. Возможность оставлять отзывы и ставить рейтинги: В Uremont пользователи могут оставлять отзывы и ставить рейтинги автосервисам. Однако количество отзывов может быть ограниченным.

5. Качество отзывов и рейтингов: у сервиса нет гарантий качества отзывов и рейтингов.

6. Сравнение рейтингов: В Uremont нет возможности сравнивать рейтинги нескольких автосервисов одновременно, что может затруднить выбор между различными вариантами.

7. Геолокационные функции: Uremont предоставляет геолокационные функции для быстрого поиска ближайших автосервисов. Однако точность определения местоположения может быть недостаточной.

Лучшие-автосервисы:

1. Удобство использования: Сервис "Лучшие-автосервисы" обычно имеет простой и интуитивно понятный интерфейс, что делает его удобным для использования даже для менее опытных пользователей.

2. Доступность и актуальность данных: Данные в "Лучших-автосервисах" обычно обновляются регулярно, однако доступность и актуальность данных может зависеть от региона и количества пользователей.

3. Возможности фильтрации и сортировки: Сервис предоставляет возможности для фильтрации и сортировки автосервисов по различным параметрам, что облегчает поиск подходящего варианта.

4. Возможность оставлять отзывы и ставить рейтинги: Пользователи имеют возможность оставлять отзывы и ставить рейтинги автосервисам. Однако может отсутствовать механизм модерации отзывов, что может привести к появлению недостоверной информации.

5. Качество отзывов и рейтингов: Качество отзывов и рейтингов не гарантируется

6. Сравнение рейтингов: Сервис не позволяет сравнивать рейтинги нескольких автосервисов одновременно, что затрудняет выбор между различными вариантами.

7. Геолокационные функции: "Лучшие-автосервисы" обычно предоставляют геолокационные функции для быстрого поиска ближайших автосервисов. Однако точность определения местоположения может быть недостаточной.

Autoreshenie:

1. Удобство использования: Autoreshenie обычно предоставляет простой и интуитивно понятный интерфейс, который удобен для пользователей всех уровней опыта.

2. Доступность и актуальность данных: Данные в Autoreshenie обновляются регулярно, что обеспечивает их актуальность. Однако доступность данных может быть ограничена в некоторых регионах.

3. Возможности фильтрации и сортировки: Сервис обычно предоставляет возможности для фильтрации и сортировки автосервисов по различным критериям, что облегчает поиск подходящего варианта.

4. Возможность оставлять отзывы и ставить рейтинги: Пользователи имеют возможность оставлять отзывы и ставить рейтинги автосервисам. Однако могут отсутствовать механизмы модерации отзывов, что может привести к появлению недостоверной информации.

5. Качество отзывов и рейтингов: Качество отзывов и рейтингов спорно.

6. Сравнение рейтингов: Сервис так же не предоставляет возможность сравнивать рейтинги нескольких автосервисов одновременно.

7. Геолокационные функции: Autoreshenie обычно предоставляет геолокационные функции для быстрого поиска ближайших автосервисов. Однако точность определения местоположения может быть недостаточной.

Likengo:

1. Удобство использования: Likengo обычно имеет простой и удобный интерфейс, который легко использовать для поиска информации об автосервисах.

2. Доступность и актуальность данных: Данные в Likengo обновляются регулярно, что обеспечивает их актуальность. Однако некоторые регионы могут иметь ограниченный объем доступных данных.

3. Возможности фильтрации и сортировки: Сервис предоставляет возможности для фильтрации и сортировки автосервисов по различным критериям, что упрощает поиск подходящего варианта.

4. Возможность оставлять отзывы и ставить рейтинги: Пользователи могут оставлять отзывы и ставить рейтинги автосервисам. Однако недостаточная модерация отзывов может привести к появлению недостоверной информации.

5. Качество отзывов и рейтингов: Качество отзывов и рейтингов под вопросом.

6. Сравнение рейтингов: Пользователи не могут сравнивать рейтинги нескольких автосервисов.

7. Геолокационные функции: Likengo предоставляет геолокационные функции для быстрого поиска ближайших автосервисов. Точность определения местоположения обычно хорошая, но в некоторых случаях может быть недостаточной.

# 1.5.3 Таблица результатов анализа аналогов

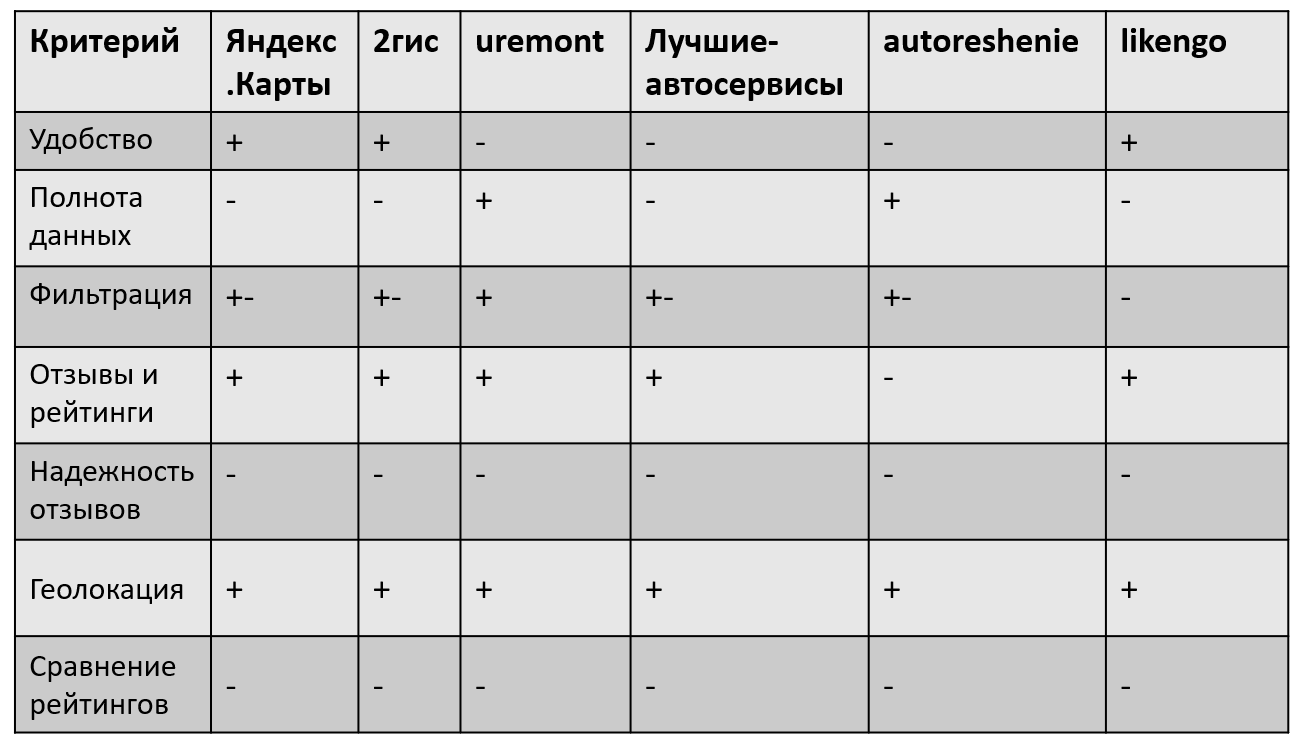


Рисунок 1 – Таблица результатов анализа аналогов

Как можно увидеть из таблицы результатов анализа аналогов, ни один из рассмотренных сервисов не обеспечивает должного уровня надежности отзывов и рейтингов. Существующие платформы не предлагают пользователям достаточной уверенности в подлинности и объективности предоставляемых данных о качестве услуг, что подрывает доверие к этим системам. Более того, ни один из аналогов не обладает встроенной функцией для сравнения информации, рейтингов и отзывов автосервисов. Это означает, что пользователи лишены возможности удобно и быстро сопоставлять различные автосервисы по ключевым показателям, что существенно усложняет процесс выбора подходящего автосервиса. В результате пользователям приходится самостоятельно искать и сопоставлять данные из различных источников, что может быть трудоемким и не всегда надежным процессом.

Эти недостатки указывают на значительные возможности для улучшения существующих решений. Разработка платформы, которая будет обеспечивать высокую надежность и объективность отзывов и рейтингов, а также предоставлять удобные инструменты для сравнения различных автосервисов, сможет удовлетворить потребности пользователей гораздо лучше. Такой подход позволит не только повысить доверие пользователей к платформе, но и сделает процесс выбора автосервиса более простым и эффективным. Внедрение данных функций станет конкурентным преимуществом, значительно выделяющим новый сервис среди существующих аналогов на рынке.

# 2 Предлагаемый способ решения поставленной задачи

# 2.1 Анализ инструментов разработки

Для создания веб-сервиса для просмотра и сравнения отзывов и рейтингов автосервисов по городам России необходимо проанализировать несколько современных технологий и выбрать более подходящие.

Технологии, выбранные для backend:

- Node.js.

Node.js – это среда выполнения JavaScript на серверной стороне. Она позволяет разрабатывать быстрые и масштабируемые серверные приложения. Node.js использует событийно-ориентированную, неблокирующую модель ввода-вывода, что делает его идеальным для приложений с интенсивным использованием данных в режиме реального времени.

Преимущества Node.js:

- высокая производительность благодаря асинхронной модели;

- один язык для фронтенда и бэкенда – JavaScript;

- большая экосистема модулей и библиотек (npm).

Сравнение с аналогами:

- Python (с Flask или Django): Python более понятен и удобен для новичков, но в реальном времени и производительности может уступать Node.js;

- Ruby on Rails: Хорошо подходит для быстрого создания прототипов, но может быть медленнее и менее масштабируемым по сравнению с Node.js.

PostgreSQL – это объектно-реляционная система управления базами данных (СУБД), известная своей надежностью, функциональностью и поддержкой сложных запросов.

Преимущества PostgreSQL:

- поддержка ACID-транзакций, что обеспечивает надежность данных;

- расширенные возможности работы с данными: индексы, полнотекстовый поиск, JSON;

- высокая производительность и масштабируемость.

Сравнение с аналогами:

- MySQL: широко используется и прост в настройке, но PostgreSQL предлагает более продвинутые функции и лучше справляется со сложными запросами;

- MongoDB: отлично подходит для неструктурированных данных и высокой скорости вставки данных, но PostgreSQL выигрывает в плане реляционных возможностей и надежности.

Sequelize ORM

Sequelize – это ORM (Object-Relational Mapping) для Node.js, который позволяет работать с SQL базами данных с использованием объектно-ориентированного подхода.

Преимущества Sequelize:

- Поддержка различных СУБД (PostgreSQL, MySQL, SQLite и др.).

- Легкость в использовании и настройке.

- Автоматическое создание и синхронизация схемы базы данных.

Сравнение с аналогами:

- TypeORM: поддерживает большее количество баз данных и имеет лучшую типизацию для TypeScript, но Sequelize более прост в освоении.

- Prisma: более современный и мощный инструмент, но может быть сложнее в настройке и использовании.

JWT-токены

JWT (JSON Web Tokens) – это стандарт для создания токенов доступа, используемых для аутентификации и авторизации.

Преимущества JWT:

- легкость в использовании и интеграции;

- безопасная передача информации между клиентом и сервером;

- масштабируемость благодаря самодостаточным токенам.

Сравнение с аналогами:

- OAuth: Более сложный и мощный стандарт, часто используется для интеграции с внешними сервисами;

- Session-based authentication: Традиционный подход с использованием серверных сессий, более простой, но менее масштабируемый по сравнению с JWT.

Технологии, выбранные для frontend:

React.js – это библиотека для создания пользовательских интерфейсов, разработанная Facebook. Она позволяет создавать динамичные и интерактивные веб-приложения с использованием компонентного подхода.

Преимущества React.js:

- высокая производительность благодаря виртуальному DOM;

- удобство разработки благодаря компонентному подходу;

- большая экосистема и сообщество.

Сравнение с аналогами:

- Vue.js: Более простой и гибкий, отлично подходит для небольших проектов, но React.js лучше поддерживается крупными компаниями;

- Angular: Полноценный фреймворк с множеством встроенных функций, но может быть сложнее в освоении и требует больше ресурсов.

Axios – это библиотека для выполнения HTTP-запросов. Она проста в использовании и поддерживает все современные функции для работы с запросами.

Преимущества Axios:

- простота в использовании и настройке;

- поддержка промисов и асинхронных операций;

- автоматическое преобразование данных в формат JSON.

Сравнение с аналогами:

- Fetch API: Встроенный в браузеры и не требует установки, но Axios предлагает больше функциональных возможностей и удобство использования;

- jQuery AJAX: Устаревающий метод, более громоздкий и менее удобный по сравнению с Axios.

React Router DOM – это библиотека для маршрутизации в React-приложениях. Она позволяет создавать многостраничные приложения с динамической навигацией.

Преимущества React Router DOM:

- простота в использовании и интеграции с React;

- поддержка как декларативного, так и императивного подходов;

- возможность создания сложных маршрутов и вложенных маршрутизаций.

Сравнение с аналогами:

- Next.js: Фреймворк для серверного рендеринга React-приложений с поддержкой маршрутизации, но может быть избыточным для простых проектов;

- Reach Router: Легкая альтернатива с более простой и удобной API, но менее популярна и имеет меньше возможностей по сравнению с React Router DOM.

MobX – это библиотека для управления состоянием в React-приложениях. Она обеспечивает простой и реактивный способ работы с состоянием.

Преимущества MobX:

- легкость в использовании и настройке;

- реактивное обновление компонентов при изменении состояния;

- простота в интеграции с существующими проектами.

Сравнение с аналогами:

- Redux: более сложный и формализованный подход к управлению состоянием, подходит для больших проектов с сложной логикой, но может быть громоздким для небольших приложений;

- Context API: Встроенный инструмент React, простой в использовании для небольших приложений, но может быть менее удобным для управления сложным состоянием по сравнению с MobX.

Эти технологии обеспечивают максимальную производительность, масштабируемость, удобство в разработке, и, соответственно, качество итогового веб-сервиса.

# 2.2 Обоснования выбора инструментов разработки

После детального изучения всех технологий и их аналогов, становится ясно, почему этот набор является наилучшим решением для выполнения поставленной задачи.

Node.js я выбрал для серверной части веб-сервиса по нескольким причинам. Во-первых, это популярная платформа с большим сообществом разработчиков и обширной экосистемой модулей и пакетов. Node.js использует асинхронную модель ввода-вывода, что позволяет обрабатывать множество запросов одновременно, делая веб-сервис более масштабируемым и производительным. Также, поскольку Node.js использует JavaScript, мы можем писать как серверную, так и клиентскую часть на одном языке, что упрощает разработку и поддержку кода.

Для управления базой данных я выбрал PostgreSQL. Это мощная реляционная база данных с открытым исходным кодом, которая известна своей надежностью, производительностью и расширяемостью. PostgreSQL поддерживает сложные запросы, транзакции и индексирование, что делает ее идеальным выбором для нашего веб-сервиса, который будет обрабатывать большое количество данных и запросов.

Sequelize ORM был выбран для упрощения взаимодействия с базой данных. Он позволяет работать с базой данных через объектно-ориентированный подход, что упрощает написание и чтение кода. С помощью Sequelize можно легко синхронизировать модели с базой данных, создавать и управлять отношениями между таблицами, а также выполнять сложные запросы без необходимости писать SQL-код вручную.

Для обеспечения безопасности веб-сервиса и управления сессиями пользователей я выбрал JWT (JSON Web Tokens). JWT-токены позволяют безопасно передавать информацию между клиентом и сервером. Они просты в использовании и позволяют легко реализовать аутентификацию и авторизацию пользователей. Токены могут храниться на клиенте, что снижает нагрузку на сервер и упрощает масштабирование приложения.

React.js был выбран для разработки пользовательского интерфейса. Это популярная библиотека для создания динамичных и интерактивных интерфейсов с использованием компонентного подхода. React.js позволяет создавать переиспользуемые компоненты, что ускоряет разработку и упрощает поддержку кода. Кроме того, React.js обеспечивает высокую производительность за счет виртуального DOM, который минимизирует количество манипуляций с реальным DOM.

Для выполнения HTTP-запросов я выбрал Axios. Это удобная библиотека для отправки и получения данных с сервера. Axios поддерживает промисы, что упрощает работу с асинхронными запросами, и предоставляет удобные методы для настройки запросов и обработки ответов. С помощью Axios мы можем легко взаимодействовать с сервером и передавать данные между клиентской и серверной частями веб-сервиса.

React Router DOM был выбран для организации маршрутизации в приложении. Он позволяет создавать динамические маршруты и управлять навигацией между страницами. Это упрощает структуру приложения и делает его более удобным для пользователей. React Router DOM интегрируется с React.js, что позволяет легко создавать многостраничные приложения с плавной навигацией.

Для управления состоянием приложения я выбрал MobX. Это библиотека, которая позволяет создавать реактивные состояния, автоматически обновляющие интерфейс при изменении данных. MobX упрощает работу с состоянием в больших приложениях и делает код более читаемым и поддерживаемым. С помощью MobX мы можем легко отслеживать изменения состояния и реагировать на них, что улучшает пользовательский опыт.

Все выбранные инструменты и технологии прошли тщательный анализ и оценку перед тем, как были окончательно утверждены для использования в разработке веб-сервиса. Этот процесс включал несколько этапов: исследование современных тенденций и инноваций в области веб-разработки, сравнение различных инструментов и технологий по их функциональности, надежности и эффективности, а также проведение тестов для оценки их производительности в реальных условиях.

На этапе исследования учитывались такие аспекты, как популярность и поддержка со стороны сообщества разработчиков, наличие документации и обучающих материалов, а также возможность интеграции с другими системами и платформами.

Выбор инструментов и технологий также был обусловлен их способностью обеспечить лучший пользовательский опыт. Это включает в себя высокую скорость загрузки страниц, удобный и интуитивно понятный интерфейс, а также надежность и безопасность работы веб-сервиса.

Таким образом, благодаря тщательно продуманному выбору инструментов и технологий, разработка веб-сервиса проходит максимально эффективно, а конечный продукт отвечает самым высоким требованиям пользователей, обеспечивая им положительный опыт взаимодействия с веб-сервисом.

# Выводы

Во второй главе описаны основные элементы создаваемого веб-сервиса для просмотра и сравнения информации, отзывов и рейтингов автосервисов по городам России, объяснена их важность и причины выбора на базе сравнения с имеющимися сервисами. Описано содержание всех страниц веб-сервиса, а также его функционал. Благодаря проведенному анализу были выявлены основные задачи разработки и наиболее важные ее аспекты.

Благодаря подробному описанию компонентов разрабатываемого веб-сервиса можно выявить более точные требования к нему и предположить ожидания пользователей. Это является одним из самых важных аспектов работы над качественным продуктом, который будет востребованным и конкурентоспособным. Его функционал не должен уступать существующим продуктам, а кроме этого, должен иметь ряд преимуществ и решать существующие в данной области проблемы.

Особенно важно уделить внимание и техническим аспектам – сохранению данных в безопасности, практичности дизайна, доступности и удобности, чтобы пользователи могли использовать веб-сервис в любых условиях и быть уверенными в его стабильной, безопасной и качественной работе. Использование современных технологий, таких как React.js для клиентской части, Node.js и Express для серверной части, а также PostgreSQL для базы данных, обеспечит высокую производительность, надежность и масштабируемость системы. Применение JWT-токенов для аутентификации повысит безопасность пользовательских данных, а интеграция с внешними API позволит поддерживать актуальность и полноту информации.

Надежная архитектура, продуманный функционал и удобный интерфейс сделают веб-сервис конкурентоспособным и привлекательным для широкой аудитории пользователей, стремящихся к качественному и безопасному обслуживанию своих автомобилей.

# 3 Архитектура, функциональная реализация и тестирование мобильного приложения

# 3.1 Архитектура веб-сервиса

Архитектура веб-сервиса должна обеспечивать высокую производительность, масштабируемость, безопасность и удобство использования. Компонентами архитектуры являются: клиентская часть (Frontend), серверная часть (Backend) и база данных.

# 3.2 Требования к функциональным характеристикам

Разрабатываемый веб-сервис должен обеспечить выполнение следующих функций:

1. Регистрация и авторизация пользователей.

Веб-сервис должен предоставлять возможность пользователям регистрироваться и авторизовываться. Это включает в себя создание нового аккаунта, вход в систему с использованием электронного адреса и пароля, восстановление пароля и возможность выхода из системы. При регистрации необходимо предусмотреть подтверждение электронной почты для повышения безопасности.

1. Поиск автосервисов.

Веб-сервис должен предоставлять функционал поиска автосервисов по различным критериям. Это включает в себя:

* поиск по названию автосервиса;
* поиск по географическому расположению;
* поиск по типу услуг.
* фильтрация результатов по различным параметрам.

1. Отображение информации об автосервисах.

Каждый автосервис должен иметь собственную страницу, на которой отображается следующая информация:

* название и адрес автосервиса;
* контактная информация;
* описание предоставляемых услуг;
* рабочие часы;
* фотографии и видео (если предоставлены);
* рейтинг и отзывы пользователей.

1. Оставление отзывов и рейтингов.

Зарегистрированные пользователи должны иметь возможность оставлять отзывы и ставить рейтинги автосервисам. Отзывы могут включать текстовый комментарий и оценку по нескольким критериям. Рейтинги должны отображаться в виде звездочек или другой графической формы.

1. Модерация отзывов.

Для обеспечения качества контента необходимо предусмотреть систему модерации отзывов. Это может включать автоматические фильтры для выявления оскорбительных или спам-сообщений, а также возможность пользователям жаловаться на неподобающие отзывы.

1. Сравнение автосервисов.

Пользователи должны иметь возможность сравнивать несколько автосервисов по различным параметрам. Это включает в себя:

* сравнение рейтингов;
* сравнение отзывов;
* сравнение предоставляемых услуг и цен;
* сравнение расстояния до автосервисов.

7.Административная панель.

Для управления контентом и пользователями веб-сервиса необходима административная панель. Администраторы должны иметь возможность:

* модерировать отзывы и рейтинги;
* добавлять и редактировать информацию об автосервисах.

Эти функциональные требования призваны обеспечить удобство и безопасность веб-сервиса.

# 3.3 Требования к нефункциональным характеристикам

Нефункциональные требования заключаются в следующем:

1. Веб-сервис должен обеспечивать безопасное хранение данных пользователей. Для этого необходимо использовать современные технологии шифрования информации и безопасные протоколы передачи данных, а также регулярно проверять сервис на уязвимости и улучшать защиту информации в соответствии с передовыми методами.
2. Веб-сервис должен корректно выполнять свои функции, обеспечивая ожидаемое поведение всех компонентов. Качественная работа функций гарантирует положительный опыт пользователей.
3. Веб-сервис должен оптимально использовать ресурсы устройства пользователя, минимизируя нагрузку на него и оптимизируя программный код. Передаваться и загружаться должны только необходимые данные.
4. В работе веб-сервиса не должно быть ошибок и сбоев, а сам сервис должен быть тщательно протестирован. Необходимо постоянно мониторить состояние сервиса во время работы, чтобы быстро реагировать на проблемы и устранять их.
5. Веб-сервис должен быть доступен для пользователей в любое время, независимо от местоположения и времени суток.
6. Пользователи должны иметь возможность получить доступ к веб-сервису через любой современный браузер.
7. Веб-сервис должен быть спроектирован таким образом, чтобы выдерживать высокую нагрузку и обеспечивать быструю работу даже при большом количестве пользователей. Это включает оптимизацию кода, использование кеширования и возможность горизонтального масштабирования.

# 3.4 Варианты использования разрабатываемого веб-сервиса

Диаграмма вариантов использования представлена на рисунке 2.

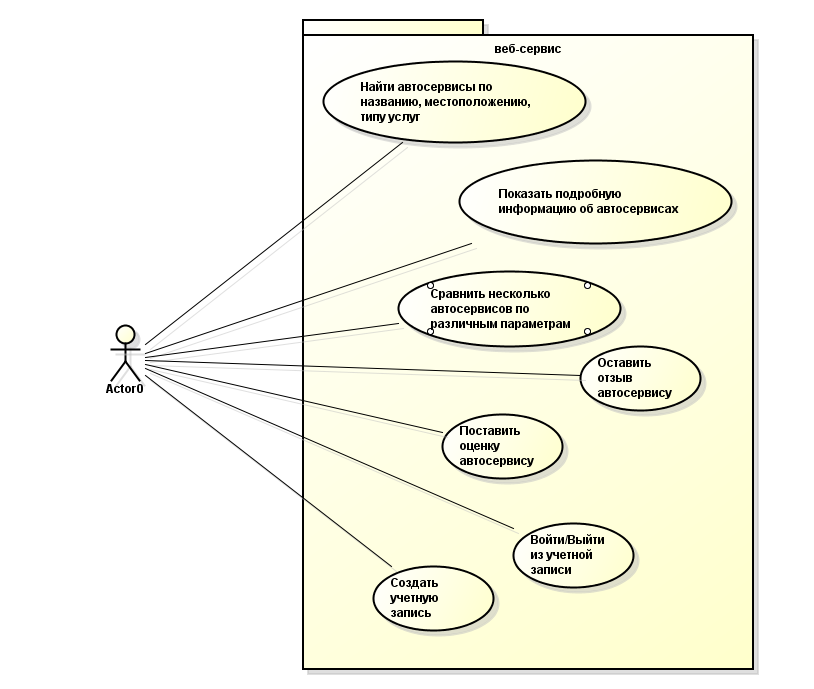


Рисунок 2 – Диаграмма вариантов использования веб-сервиса

Сценарий регистрации пользователя:

* пользователь переходит на страницу регистрации;
* пользователь вводит свои данные в поля для логина и пароля;
* пользователь подтверждает регистрацию.

Сценарий авторизации пользователя:

* пользователь переходит на страницу входа;
* пользователь вводит свои данные в поля для логина и пароля;
* пользователь подтверждает вход.

Сценарий выставления оценки автосервису:

* пользователь переходит на страницу автосервиса;
* пользователь нажимает на желаемую оценку.

Сценарий оставления отзыва автосервису:

* пользователь переходит на страницу автосервиса;
* пользователь нажимает кнопку оставления отзыва;
* пользователь пишет отзыв;
* пользователь подтверждает отправку отзыва.

Сценарий просмотра подробной информации об автосервисе записи:

* пользователь переходит на страницу со списком автосервисов;
* пользователь нажимает на интересующий его автосервис;
* пользователю открывается страница с подробной информацией об автосервисе.

Сценарий поиска автосервиса:

* пользователь переходит на страницу со списком веб-сервисов;
* пользователь вводит ключевые слова в поисковой строке;
* пользователю покажутся все автосервисы, удовлетворяющие поисковому запросу.

Сценарий фильтрации списка автосервисов:

* пользователь переходит на страницу со списком автосервисов
* пользователь выбирает желаемые категории, по которым произойдет фильтрация
* пользователю покажутся все автосервисы, удовлетворяющие выбранным фильтрам

# 3.5 Проектирование веб-сервиса

Устройство веб-сервиса представлено на рисунке 3.

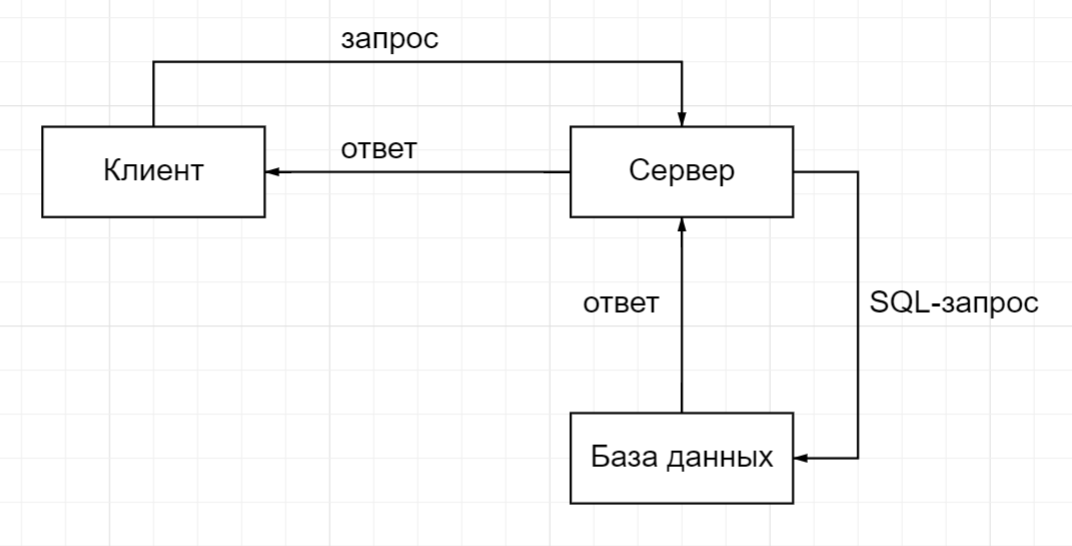


Рисунок 3 – Устройство веб-сервиса

Веб-сервис состоит из трёх основных компонентов: клиентской части, серверной части и базы данных. Эти компоненты взаимосвязаны и взаимодействуют друг с другом, чтобы обеспечить пользователям удобный и безопасный доступ к данным.

Клиентская часть веб-сервиса отвечает за взаимодействие с пользователем. Она включает в себя пользовательский интерфейс и логику взаимодействия с серверной частью. В данном проекте для клиентской части используется React.js, популярная библиотека для создания пользовательских интерфейсов. С помощью React.js можно создавать динамичные, отзывчивые и легко масштабируемые пользовательские интерфейсы.

Основными задачами на стороне клиента являются:

1. Отображение данных: Клиентская часть отвечает за получение и отображение данных со стороны сервера, таких как списки автомобильных услуг, рейтинги и отзывы.

2 Взаимодействие с пользователем: Обработка действий пользователя, таких как фильтрация, сортировка, добавление оценок и отзывов.

3 Отправка запросов на сервер: когда необходимо получить или отправить данные, клиентская сторона отправляет HTTP-запросы на серверную сторону.

Серверная сторона обрабатывает запросы от клиентской стороны и взаимодействует с базой данных. Для реализации серверной части используется Node.js с фреймворком Express для создания быстрых и масштабируемых серверных приложений.

Основными задачами серверной части являются:

1. Обработка запросов: Сервер получает запросы от клиента, обрабатывает их и возвращает запрошенные данные или подтверждение.

2 Аутентификация и авторизация: Использование JWT (JSON Web Tokens) для защиты и контроля доступа к различным функциям веб-сервиса.

3 Логика приложения: Обработка бизнес-логики, например подсчет рейтинга, обработка отзывов и управление пользователями.

4 Взаимодействие с базой данных: Выполнение запросов к базе данных для чтения и записи данных.

В базе данных хранится вся информация, необходимая для работы веб-сервиса. В данном проекте используется PostgreSQL, мощная и надежная реляционная СУБД, позволяющая хранить и управлять данными.

Основными задачами базы данных являются:

1. Хранение данных: Хранение информации об автосервисах, пользователях, рейтингах и отзывах.

2. Целостность данных: Обеспечение постоянства и единообразия данных.

3. обработка запросов: Выполнение SQL-запросов, поступающих со стороны сервера, для получения, вставки, обновления и удаления данных.

Взаимодействие компонентов

1. Сторона клиента и сторона сервера:

- пользователь на стороне клиента выполняет действие, например, вводит название города и нажимает кнопку «Поиск»;

- клиентская часть формирует HTTP-запрос (например, GET или POST) и отправляет его на серверную часть;

- серверная часть получает запрос, обрабатывает его и генерирует ответ, который затем отправляется обратно клиенту;

- клиентская часть получает ответ и обновляет интерфейс в соответствии с полученными данными.

2. Серверная часть и база данных:

- для запроса, требующего данных из базы данных (например, список автосервисов в определенном городе), серверная часть отправляет соответствующий SQL-запрос в базу данных;

- база данных обрабатывает запрос и возвращает результаты (например, список автосервисов с их рейтингами и отзывами) на серверную часть;

- серверная сторона создает ответ на запрос клиента, включая данные, полученные из базы данных, и отправляет его обратно клиентской стороне.

Важно понимать роль каждого компонента:

- клиентская часть предоставляет пользователю удобный и интуитивно понятный интерфейс для взаимодействия с веб-сервисом;

- серверная часть служит посредником между клиентом и базой данных, обрабатывает запросы, выполняет бизнес-логику и обеспечивает безопасность данных;

- база данных отвечает за безопасное хранение и управление данными и обеспечивает быстрый доступ к необходимой информации.

Такая архитектура обеспечивает высокую производительность, безопасность и масштабируемость веб-сервиса и позволяет пользователям эффективно искать и сравнивать автосервисы по городам России.

# 3.6 Проектирование базы данных

Структура базы данных представлена на рисунке 4.

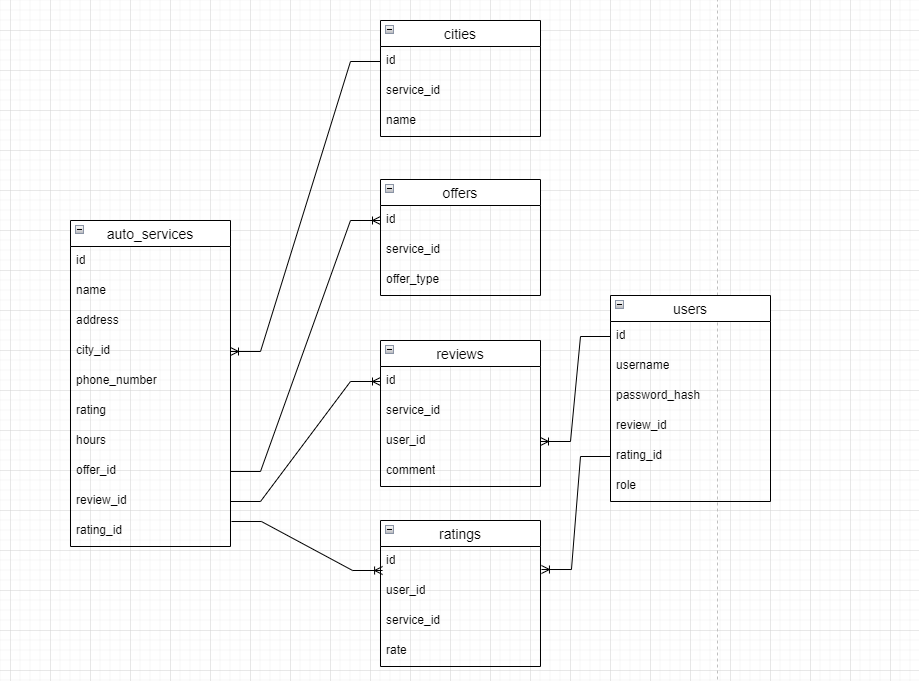


Рисунок 4 – структура базы данных концептуального уровня

Представленная база данных состоит из следующих таблиц:

1. Пользователь: хранит информацию о: имени пользователя, его пароле, роли (Администратор или Обычный пользователь), оставленном отзыве и рейтинге.
2. Автосервис: хранит информацию о названии, адресе, контактных данных, часах работы, услугах, отзывах и рейтингах автосервисов.
3. Город: хранит название и id автосервиса, который находится в этом городе.
4. Услуга: хранит id автосервиса и название типа услуги.
5. Отзыв: хранит id автосервиса, к которому он написан, id пользователя, которым он написан, и само свое содержание.
6. Рейтинг: хранит хранит id автосервиса, которому он поставлен, id пользователя, который его поставил, и само числовое значение рейтинга.

# 3.7 Проектирование классов

Диаграмма классов концептуального уровня представлена на рисунке 5.

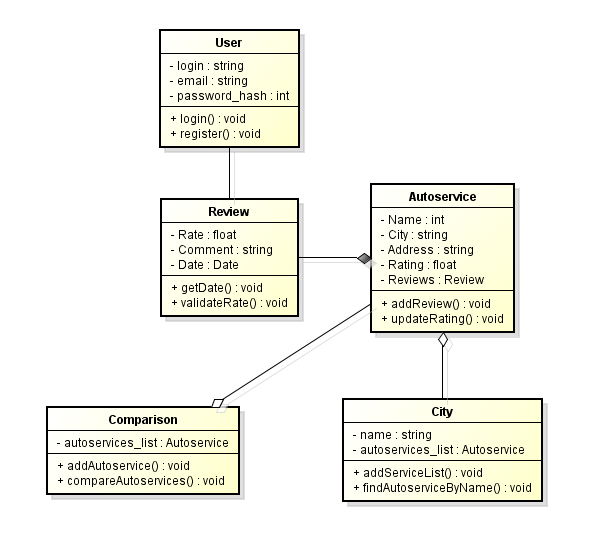


Рисунок 5 – структура классов концептуального уровня

Структура классов аналогична базе данных и включает в себя следующие классы:

Пользователь:

1. Автосервис: хранит информацию о названии, адресе, контактных данных, часах работы, услугах, отзывах и рейтингах автосервисов.
2. Город: хранит название и id автосервиса, который находится в этом городе.
3. Услуга: хранит id автосервиса и название типа услуги.
4. Отзыв: хранит id автосервиса, к которому он написан, id пользователя, которым он написан, и само свое содержание.
5. Рейтинг: хранит хранит id автосервиса, которому он поставлен, id пользователя, который его поставил, и само числовое значение рейтинга.

# 3.8 Ручное тестирование веб-сервиса

Наиболее подходящий вариант проверки работоспособности веб-сервиса – ручное тестирование по сценариям. Для тестирования функциональности системы были разработаны сценарии ручного тестирования, учтя все возможные действия пользователя. В сценариях описываются шаги тестирования и результаты их выполнения.

Тест-кейс №1. Пользователь регистрируется в системе.

Действия тестировщика:

* 1. На главной странице нажать на кнопку Зарегистрироваться.
  2. В появившиеся формы ввести корректные логин и пароль.
  3. Нажать на кнопку Зарегистрироваться.

Ожидаемый результат: перед пользователем появится страница для авторизации.

Тест-кейс №2. Пользователь авторизуется в системе.

Действия тестировщика:

* 1. На главной странице нажать на кнопку Войти.
  2. В появившиеся формы ввести корректные логин и пароль.
  3. Нажать на кнопку Войти.

Ожидаемый результат: перед пользователем появится страница профиля.

Тест-кейс №3. Пользователь ищет автосервисы без фильтров.

Действия тестировщика:

* 1. На главной странице в форме для поиска автосервисов ввести ключевые слова.
  2. Нажать на кнопку Поиск.

Ожидаемый результат: перед пользователем появится страница со списком автосервисов, удовлетворяющим введенному запросу.

Тест-кейс №4. Пользователь ищет автосервисы с фильтрами.

Действия тестировщика:

* 1. На главной странице в форме для поиска автосервисов ввести ключевые слова.
  2. Нажать на кнопку Фильтры.
  3. Выбрать желаемые критерии, по которым будет осуществляться фильтрация автосервисов.
  4. Нажать на кнопку Поиск.

Ожидаемый результат: перед пользователем появится страница со списком автосервисов, удовлетворяющим введенному запросу и выбранным фильтрам.

Тест-кейс №5. Пользователь выбирает список всех автосервисов.

Действия тестировщика:

* + 1. На главной странице нажать на кнопку Поиск.

Ожидаемый результат: перед пользователем появится страница со списком всех автосервисов.

Тест-кейс №6. Пользователь выбирает один автосервис из списка.

Действия тестировщика:

* 1. На странице со списком автосервисов нажать на любую карточку автосервиса.

Ожидаемый результат: перед пользователем появится страница с информацией об автосервисе.

Тест-кейс №7. Авторизованный пользователь пишет отзыв на автосервис.

Действия тестировщика:

* 1. На странице с информацией об автосервисе нажать на кнопку Оставить отзыв.
  2. На странице написания отзыва ввести текст отзыва в форму.
  3. Нажать на кнопку Отправить.

Ожидаемый результат: на странице автосервиса появится написанный пользователем отзыв.

Тест-кейс №8. Неавторизованный пользователь пытается написать отзыв на автосервис.

Действия тестировщика:

* 1. На странице с информацией об автосервисе нажать на кнопку Оставить отзыв.

Ожидаемый результат: перед пользователем появится сообщение о том, что необходимо авторизоваться.

Тест-кейс №9. Пользователь пытается авторизоваться с неверным логином.

Действия тестировщика:

* 1. На странице авторизации ввести неверный логин.

Ожидаемый результат: перед пользователем появится сообщение о том, логин не существует.

Тест-кейс №10. Пользователь пытается авторизоваться с неверным паролем.

Действия тестировщика:

* 1. На странице авторизации ввести неверный пароль.

Ожидаемый результат: перед пользователем появится сообщение о том, пароль не подходит.

Тест-кейс №11. Пользователь ставит оценку автосервису.

Действия тестировщика:

1. На странице информации об автосервисе нажать на любую из пяти звезд.

Ожидаемый результат: на странице информации об автосервисе появится выставленная оценка.

Тест-кейс №12. Пользователь сравнивает автосервисы.

Действия тестировщика:

* 1. На странице со списком автосервисов нажать на кнопку сравнить автосервисы.
  2. Выбрать 10 автосервисов.

Ожидаемый результат: пользователю выведется таблица с результатом сравнения выбранных автосервисов.

Тест-кейс №13. Пользователь выбирает город.

Действия тестировщика:

* 1. На главной странице нажать кнопку Выберите город.
  2. Выбрать нужный город.

Ожидаемый результат: на главной странице будет отображаться выбранный город.

# Выводы

В третьей главе рассмотрены архитектура веб-сервиса, реализация его функциональных возможностей и процесс тестирования. Основные выводы следующие:

Архитектура веб-сервиса разработана для обеспечения высокой производительности, масштабируемости и безопасности. Основные компоненты включают клиентскую часть на React.js и серверную часть с использованием Node.js, Express и PostgreSQL. Взаимодействие через RESTful API обеспечивает гибкость и надежность системы. JWT-токены используются для аутентификации и авторизации, что улучшает безопасность и управляемость пользовательскими сессиями. Sequelize ORM облегчает взаимодействие с базой данных, делая код более чистым и поддерживаемым.

Веб-сервис показал стабильную работу при различных нагрузках, корректное функционирование на различных устройствах и ОС, а также высокий уровень безопасности данных. Тщательное тестирование обеспечило выявление и устранение дефектов на ранних стадиях разработки, гарантируя высокое качество конечного продукта. Проведенные тесты включали как автоматическое тестирование функционала, так и юзабилити-тестирование с участием реальных пользователей, что позволило сделать интерфейс максимально удобным и интуитивно понятным.

Разработка архитектуры, реализация функциональных возможностей и комплексное тестирование обеспечили создание высококачественного, надежного и безопасного веб-сервиса для просмотра и сравнения информации, отзывов и рейтингов автосервисов по городам России. Веб-сервис удовлетворяет современные требования пользователей и способствует улучшению их опыта взаимодействия с автосервисами, повышая их осведомленность и уверенность в выборе услуг.

# 4 Оценка и результаты эффективности мобильного приложения

# 4.1 Методы оценки эффективности мобильного приложения

Оценка эффективности веб-сервиса имеет большое значение для его успешного функционирования и развития. Она позволяет получить полное представление о взаимодействии пользователей с сервисом, их ожиданиях и потребностях, а также выявить проблемные области и возможности для улучшения. Включение этого раздела в дипломную работу демонстрирует понимание важности и методов оценки, а также обосновывает выбор подходов и инструментов, используемых в проекте.

Оценка эффективности помогает понять, насколько хорошо веб-сервис выполняет свои задачи и достигает поставленных целей. Она включает анализ таких показателей, как удобство использования, производительность, удовлетворённость пользователей, техническая надёжность и конкурентоспособность. Важность оценки эффективности заключается в предоставлении данных для принятия обоснованных решений по улучшению сервиса.

Одна из ключевых целей оценки — понимание пользовательского опыта. Это подразумевает анализ взаимодействия пользователей с интерфейсом, лёгкости поиска нужной информации и выполнения необходимых действий. Изучение путей пользователей, тепловых карт и показателей отказов помогает выявить проблемы в навигации и структуре сайта, которые могут мешать пользователям. Это позволяет внести изменения для улучшения удобства использования и повышения удовлетворённости пользователей.

Оценка эффективности также включает анализ производительности веб-сервиса. Технические аспекты, такие как скорость загрузки страниц, стабильность работы сервера и отсутствие ошибок, напрямую влияют на пользовательский опыт. Автоматизированные тесты производительности и системы мониторинга ошибок поддерживают высокий уровень технической надёжности, что критически важно для удержания пользователей и повышения их лояльности.

Сбор и анализ обратной связи от пользователей позволяет понять, что им нравится или не нравится в веб-сервисе. Опросы, отзывы и комментарии предоставляют ценную информацию о реальных потребностях и ожиданиях пользователей, что помогает адаптировать сервис под их требования. Регулярное получение обратной связи и её анализ позволяют быстро реагировать на замечания и предложения, что способствует постоянному улучшению сервиса.

Конкурентный анализ — ещё один важный аспект оценки эффективности. Сравнение веб-сервиса с аналогичными продуктами на рынке позволяет выявить его сильные и слабые стороны. Это помогает понять, как улучшить функциональность и качество сервиса для успешной конкуренции и привлечения большего числа пользователей. Важный результат оценки эффективности — возможность принятия обоснованных решений по развитию веб-сервиса. На основе собранных данных можно определить приоритетные направления для улучшения, разработать и внедрить необходимые изменения, а затем снова провести оценку для подтверждения их эффективности. Таким образом, процесс оценки эффективности становится цикличным, способствуя постоянному развитию и совершенствованию веб-сервиса. Включение раздела об оценке эффективности веб-сервиса в дипломную работу демонстрирует систематический и профессиональный подход к разработке продукта. Это показывает, что разработчик понимает важность пользовательского опыта, технической надёжности и конкурентного анализа, а также умеет применять различные методы и инструменты для всесторонней оценки и принятия обоснованных решений по улучшению веб-сервиса.

Оценка эффективности веб-сервиса для сравнения и анализа отзывов и рейтингов автосервисов по городам России — это сложный процесс, включающий различные методы измерения функциональности и удовлетворённости пользователей. Один из ключевых методов — юзабилити-тестирование, где реальные пользователи взаимодействуют с сервисом, а исследователи анализируют возникающие трудности и собирают отзывы. Этот подход помогает выявить слабые места в интерфейсе и навигации, влияющих на пользовательский опыт.

Также важен сбор и анализ количественных данных с помощью инструментов веб-аналитики, таких как Google Analytics. Эти данные включают показатели трафика, время пребывания на сайте, процент отказов, конверсии и другие показатели, которые показывают эффективность привлечения и удержания пользователей.

Другой важный метод — опросы пользователей, содержащие открытые и закрытые вопросы для сбора мнений о сервисе. Опросы могут проводиться через анкеты, рассылки по электронной почте или формы обратной связи на самом сервисе. Сбор качественных данных позволяет лучше понять удовлетворённость пользователей и выявить области для улучшения.

Для оценки технической эффективности проводятся стресс- и нагрузочные тесты, определяющие поведение системы при высокой нагрузке и выявляющие узкие места в производительности.

Анализ отзывов и оценок пользователей также важен для оценки производительности. Систематический анализ содержания отзывов помогает выявить общие тенденции, проблемы и положительные стороны услуг автосервисов. Важно также оценивать репрезентативность и достоверность отзывов с помощью текстового анализа и методов выявления аномалий.

Мониторинг конкурирующих сервисов позволяет сравнивать функциональность, удобство использования и возможности для улучшения. Регулярный бенчмаркинг помогает оставаться конкурентоспособным и адаптироваться к изменениям на рынке.

Не менее важный метод — анализ показателей возврата пользователей и их лояльности. Постоянные пользователи свидетельствуют о высоком уровне удовлетворённости и доверия к сервису. Изучение их поведения даёт ценную информацию для дальнейшего развития и совершенствования платформы.

Интеграция методов оценки, таких как A/B-тестирование, позволяет экспериментировать с элементами интерфейса и выявлять наиболее эффективные решения на основе статистически значимых данных. Такой подход оптимизирует пользовательский опыт и повышает общую эффективность сервиса.

Комплексная оценка эффективности веб-сервиса предполагает использование качественных и количественных методов для всестороннего анализа пользовательского опыта, производительности системы и конкурентоспособности. Каждый метод вносит свой вклад в понимание эффективности и помогает выявить сильные стороны и области для улучшения, что в итоге приводит к созданию более качественного и удобного сервиса.

# 4.2 Опросное анкетирование пользователей по эффективности веб-сервиса

Опросное анкетирование – это метод сбора информации, который включает в себя задавание стандартных вопросов людям с помощью анкеты. В анкете могут быть разные виды вопросов: закрытые, где нужно выбрать ответ из предложенных вариантов, и открытые, где можно ответить своими словами. Закрытые вопросы удобны для анализа, а открытые помогают понять мнение респондентов более глубоко. Анкетирование можно проводить разными способами: лично, по телефону, по почте или через интернет. Каждый из этих методов имеет свои плюсы и минусы. Личное анкетирование позволяет подробно объяснить вопросы и получить более развернутые ответы, но оно занимает много времени и денег. Телефонное и почтовое анкетирование дешевле, но часто сталкивается с низкой ответной активностью. Онлайн-анкеты становятся все популярнее, так как интернет позволяет быстро и недорого охватить большое количество людей. Проведение опросного анкетирования обычно включает несколько этапов. Сначала нужно определить цели и задачи исследования, затем разработать анкету, выбрать и пригласить респондентов, провести сам опрос, обработать и проанализировать данные, а потом интерпретировать и представить результаты. Важно тщательно продумать вопросы анкеты и правильно выбрать респондентов, чтобы данные были точными и представляли интересующую нас группу людей. Опросное анкетирование широко используется в социальных и маркетинговых исследованиях, потому что оно позволяет получить количественные данные о мнениях, предпочтениях и поведении людей. Эти данные помогают принимать обоснованные решения, разрабатывать стратегии и оценивать эффективность различных программ и мероприятий.

Для выявления эффективности веб-сервиса было проведено опросное анкетирование 29 человек – владельцев автомобилей и обычных пользователей.

Далее на рисунках 6- 16 представлены результаты проведенного опроса, отражающие уровень удовлетворенности пользователей веб-сервисом для просмотра и сравнения информации, отзывов и рейтингов автосервисов.

Эти рисунки иллюстрируют мнения и впечатления пользователей относительно различных аспектов веб-сервиса, включая удобство использования, доступность информации, точность и достоверность отзывов, а также функциональность и производительность платформы.

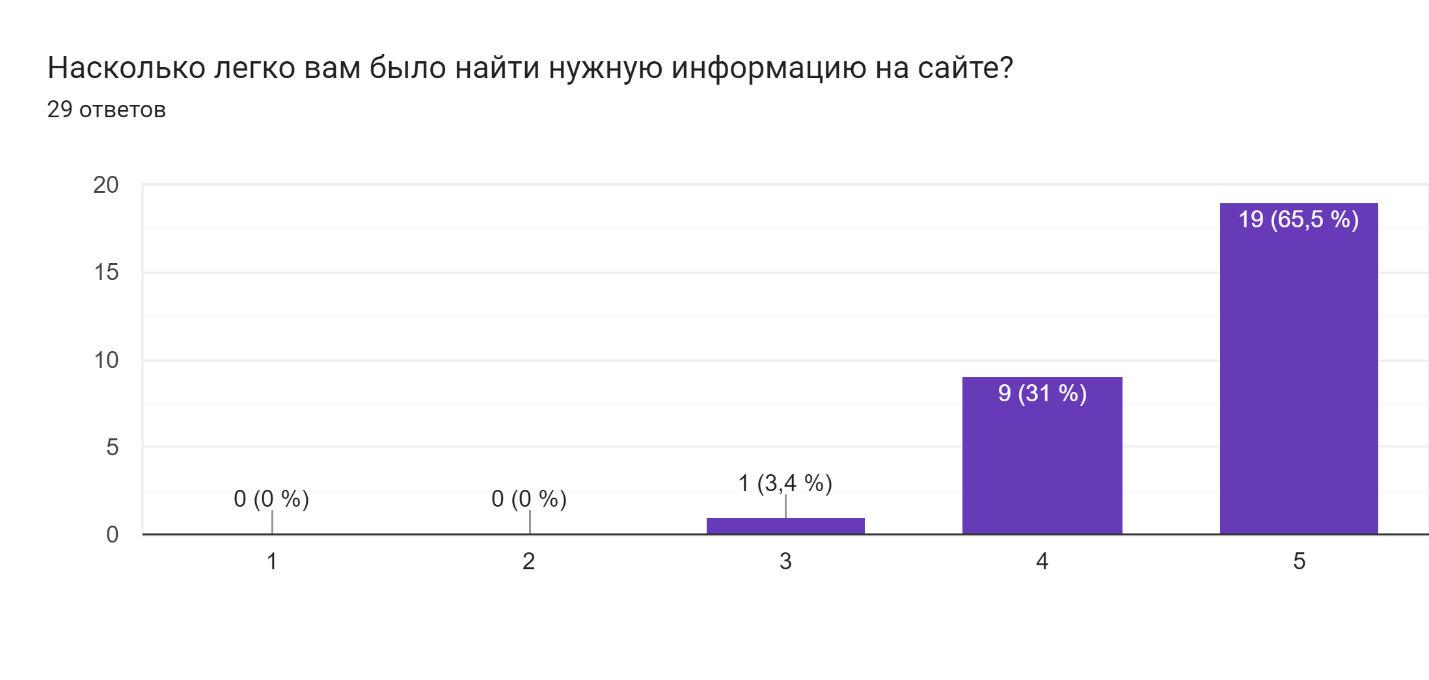


Рисунок 6 - Результаты опросной анкеты, часть 1

Диаграмма ответов в Формах. Вопрос: Были ли у вас сложности при навигации по веб-сайту?
. Количество ответов: 29 ответов.

Рисунок 7 - Результаты опросной анкеты, часть 2

Диаграмма ответов в Формах. Вопрос: Насколько понятным был интерфейс веб-сервиса?
. Количество ответов: 29 ответов.

Рисунок 8 - Результаты опросной анкеты, часть 3

Диаграмма ответов в Формах. Вопрос: Насколько приятным был дизайн веб-сервиса?
. Количество ответов: 29 ответов.

Рисунок 9- Результаты опросной анкеты, часть 4

Диаграмма ответов в Формах. Вопрос: Были ли моменты, когда вам приходилось возвращаться назад или искать помощь для выполнения задач?
. Количество ответов: 29 ответов.

Рисунок 10 - Результаты опросной анкеты, часть 5

Диаграмма ответов в Формах. Вопрос: Насколько актуальной показалась вам информация о рейтингах и отзывах автосервисов?
. Количество ответов: 29 ответов.

Рисунок 11 - Результаты опросной анкеты, часть 6

Диаграмма ответов в Формах. Вопрос: Насколько легко вам было использовать фильтры для поиска автосервисов?
. Количество ответов: 29 ответов.

Рисунок 12 - Результаты опросной анкеты, часть 7

Диаграмма ответов в Формах. Вопрос: Были ли сложности при оставлении отзыва или рейтинга?
. Количество ответов: 29 ответов.

Рисунок 13 - Результаты опросной анкеты, часть 8

Диаграмма ответов в Формах. Вопрос: Возникало ли у вас чувство, что отзывы могут быть поддельными?
. Количество ответов: 29 ответов.

Рисунок 14 - Результаты опросной анкеты, часть 9

Диаграмма ответов в Формах. Вопрос: Насколько понятным был результат сравнения рейтингов различных автосервисов?
. Количество ответов: 29 ответов.

Рисунок 15 - Результаты опросной анкеты, часть 10

Диаграмма ответов в Формах. Вопрос: Порекомендуете ли вы наш веб-сервис своим друзьям и коллегам?
. Количество ответов: 29 ответов.

Рисунок 16 - Результаты опросной анкеты, часть 11

На основе полученных данных были получены следующие результаты:

1. Насколько легко вам было найти нужную информацию на сайте?

По шкале от 1 до 5, где 1 — это трудно, а 5 – легко:

* 19 опрошенных (65,5%) выбрали 5;
* 9 опрошенных (31%) выбрали 4;
* всего 1 опрошенный (3,4%) выбрал 3.

2. Были ли у вас сложности при навигации по веб-сайту?

* 22 опрошенным (75,9%) было несложно передвигаться по сайту;
* 7 опрошенных (24,1%) испытывали трудности при навигации по веб-сервису;

3. Насколько понятным был интерфейс веб-сервиса?

По шкале от 1 до 5, где 1 — это совсем не понятный, а 5 – полностью понятный:

* 19 опрошенных (65,5%) выбрали 5;
* 9 опрошенных (31%) выбрали 4;
* всего 1 опрошенный (3,4%) выбрал 3;

4. Насколько приятным был дизайн веб-сервиса?

По шкале от 1 до 5, где 1 — это совсем не приятный, а 5 – очень приятный:

* 21 опрошенных (72,4%) выбрали 5;
* 7 опрошенных (24,1%) выбрали 4;
* И всего 1 опрошенный (3,4%) выбрал 3.

5. Были ли моменты, когда вам приходилось возвращаться назад или искать помощь для выполнения задач:

* 20 опрошенных (69%) справлялись со своими задачами самостоятельно;
* 9 опрошенных (31%) испытывали трудности при выполнении своих задач;

6. Насколько актуальной показалась вам информация о рейтингах и отзывах автосервисов?

По шкале от 1 до 5, где 1 — это совсем не актуальная, а 5 – актуальная:

* 18 опрошенных (62,1%) выбрали 5;
* 7 опрошенных (34,5%) выбрали 4;
* И всего 1 опрошенный (3,4%) выбрал 3.

7. Насколько легко вам было использовать фильтры для поиска автосервисов?

По шкале от 1 до 5, где 1 — это совсем не легко, а 5 – очень легко:

* 20 опрошенных (69%) выбрали 5;
* 7 опрошенных (27,6%) выбрали 4;
* И всего 1 опрошенный (3,4%) выбрал 3.

8. Были ли сложности при оставлении отзыва или рейтинга?

* 21 опрошенный (72,4%) без затруднений оставляли отзыв или ставили рейтинг;
* 8 опрошенных (27,6%) испытывали трудности при оставлении отзыва или рейтинга.

9. Возникало ли у вас чувство, что отзывы могут быть поддельными?

* 19 опрошенных (65,5%) были уверены в достоверности отзывов;
* 10 опрошенных (34,5%) сомневались в правдивости отзывов.

10. Насколько понятным был результат сравнения рейтингов различных автосервисов?

По шкале от 1 до 5, где 1 — это совсем не легко, а 5 – очень легко:

* 18 опрошенных (62,1%) выбрали 5;
* 9 опрошенных (31%) выбрали 4;
* всего 2 опрошенных (6,9%) выбрал 3.

11. Порекомендуете ли вы наш веб-сервис своим друзьям и коллегам?

* 23 опрошенных (79,3%) порекомендовали бы сайт знакомым;
* 5 опрошенных (17,2%) затруднялись дать ответ;
* 1 опрошенный (3,4%) не порекомендовал бы веб-сервис.

Результаты опроса пользователей позволяет выявить сильные и слабые стороны веб-сервиса, понять, насколько веб-сервис отвечает требованиям пользователей, и какие конкретно части программы требует улучшения.

Обратив внимания на диаграммы, можно увидеть, большинство пользователей довольны веб-сервисом, не испытывали затруднений и оценивают работу с ним положительно.

# Выводы

Четвертая глава охватывает методы оценки эффективности веб-сервиса для просмотра и сравнения рейтингов и отзывов автосервисов по городам России, а также анализ результатов этих оценок, направленные на определение уровня удовлетворенности пользователей. Методы оценки, такие как опросы, анкетирование, анализ отзывов и ключевые показатели эффективности, продемонстрировали свою значимость в получении объективной и всесторонней оценки веб-сервиса. Эти методы позволяют выявлять как положительные стороны, так и области, требующие улучшений.

Результаты анкетирования и анализа отзывов пользователей показали высокий уровень удовлетворенности, что свидетельствует о правильном выборе функциональных и технических решений при разработке веб-сервиса. В частности, пользователи высоко оценили интуитивность интерфейса, актуальность данных и удобство фильтрации и сортировки информации. Тем не менее, выявленные недостатки в навигации, персонализации и уровне технической поддержки указывают на необходимость дальнейших улучшений.

Проведенные тесты подтвердили, что веб-сервис не только соответствует заявленным функциональным требованиям, но и обеспечивает высокий уровень пользовательского опыта. Это достигнуто благодаря продуманной архитектуре, эффективной реализации функционала и тщательному тестированию. Веб-сервис готов к выпуску и использованию, и он сможет удовлетворить потребности пользователей, предлагая им надежный инструмент для поиска и сравнения автосервисов по городам России.

# Заключение

В дипломной работе всесторонне проанализированы аспекты разработки и внедрения веб-сервиса для отображения и сравнения рейтингов и отзывов об автосервисах в городах России. Были рассмотрены теоретические и практические аспекты создания такого программного обеспечения, что позволило глубоко понять существующие проблемы и предложить эффективные решения.

В рамках теоретического анализа изучены основные принципы оценки и классификации автосервисов, методы сбора отзывов, критерии оценки качества услуг и факторы, влияющие на удовлетворённость клиентов.

Обзор существующих веб-сервисов показал проблемы с навигацией, релевантностью данных, фильтрацией и сортировкой, а также низким уровнем доверия к рейтингам. Для решения этих проблем был выбран перспективный подход с использованием современных технологий и инструментов.

Для обеспечения высокой производительности, надёжности и безопасности веб-сервиса разработаны и реализованы функциональные требования, особое внимание уделено дизайну пользовательского интерфейса.

В результате проведённой работы создан веб-сервис для сравнения и анализа информации о рейтингах и отзывах об автосервисах в городах России, отвечающий современным требованиям и удовлетворяющий разнообразные потребности пользователей. Веб-сервис включает функции для эффективного поиска и выбора автосервисов, фильтрации и сортировки, отображения геолокационных данных и возможности оставлять и давать отзывы.

Высокий уровень удовлетворённости пользователей подтверждает правильность выбранных подходов и решений в процессе разработки веб-сервиса. Использование современных технологий и методов разработки, а также учёт отзывов пользователей позволили создать продукт высокого качества и функциональности.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

# Приложение А

# Справка о результатах проверки выпускной квалификационной работы на наличие заимствований

# Приложение Б

# Техническое задание

# Приложение В

# Руководство системного программиста