

ИНЖЕНЕРИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОНЛАЙН-ПРОКАТА АВТОМОБИЛЕЙ С ИСПОЛЬЗОВАНИЕМ VUE.JS И NODE.JS

Аннотация. В статье представлен комплексный обзор этапов разработки веб-системы проката автомобилей на базе Vue.js, Node.js и MySQL. Процесс проектирования включает анализ требований, построение архитектуры, создание базы данных, реализацию интерфейса и серверной части, интеграцию компонентов и оценку экономической эффективности выбранного технологического стека. Показано, что разработанная система обладает высокой масштабируемостью, производительностью и удобством использования, что делает её перспективным инструментом для автоматизации процессов онлайн-бронирования и управления автопарком.

Ключевые слова: Vue.js, Node.js, MySQL, веб-система, архитектура, прокат автомобилей.

Современные веб-системы требуют комплексного подхода к проектированию, включающего анализ требований, архитектурные решения, интерфейсное проектирование и обеспечение производительности. Разработка системы проката автомобилей на основе Vue.js характеризуется многоэтапностью и структурированностью, обеспечивающими последовательное достижение всех функциональных и нефункциональных целей. На первоначальном этапе осуществляется анализ требований, в рамках которого определяются ключевые функции – регистрация пользователей, бронирование, управление заказами, настройка тарифов и работа административной панели. Сбор требований выполняется путём интервьюирования потенциальных пользователей, анализа существующих аналогов и изучения рыночных тенденций. На основе собранных данных формируются пользовательские сценарии и создаётся модульная структура, включающая подсистемы управления пользователями, каталог автомобилей, модуль бронирований, модуль выставления платежей и административный раздел.

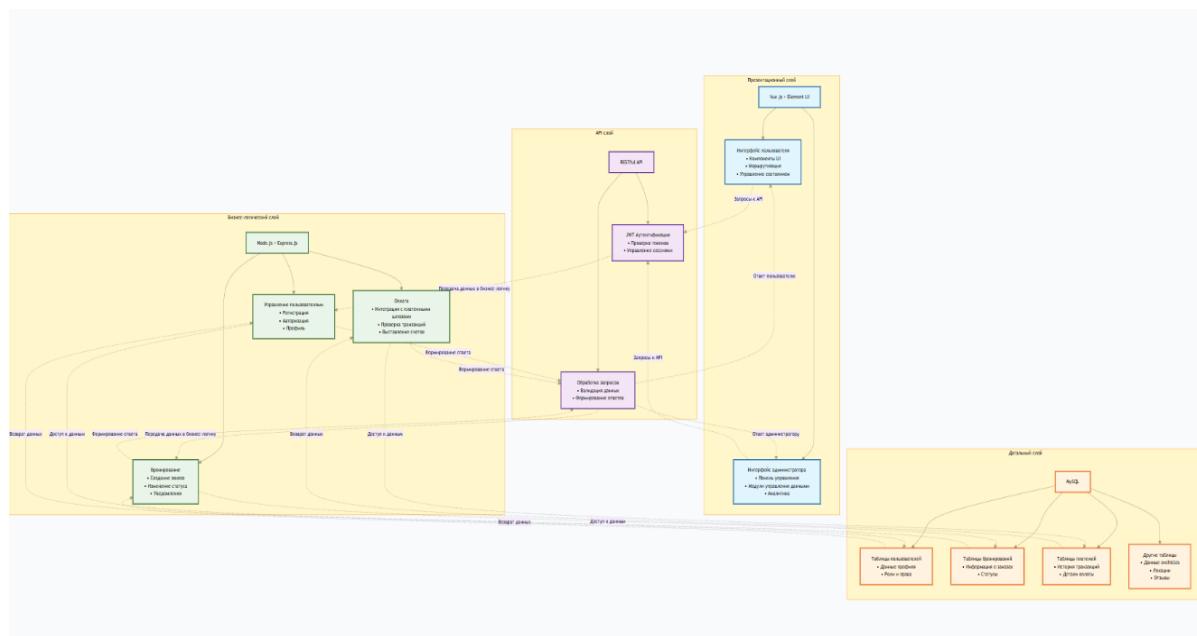


Рисунок 1. Архитектура системы проката автомобилей



Следующий этап связан с проектированием архитектуры программной системы. Формируется многоуровневая структура, состоящая из клиентской части на Vue.js 3, серверной логики на Node.js с использованием фреймворка NestJS, уровня данных на базе MySQL и API-уровня, взаимодействующего через RESTful-интерфейсы. Подобная архитектура позволяет достичь высокой производительности и масштабируемости, упрощает поддержку и обеспечивает удобную интеграцию новых модулей. Особое внимание уделяется проектированию базы данных: создаются ER-диаграммы, формируются таблицы для хранения информации о пользователях, транспортных средствах, заказах, тарифах, отзывах и платежах. Определяются типы данных, первичные и внешние ключи, индексация и хранимые процедуры, позволяющие оптимизировать операции, например расчёт стоимости заказа.

Последующим ключевым шагом является разработка пользовательского интерфейса. Использование Vue.js 3 обеспечивает гибкость компонентного подхода и позволяет создавать переиспользуемые элементы интерфейса. Разрабатывается единая дизайн-система, включающая цветовые решения, шрифты и UI-компоненты. Страницы интерфейса проектируются с учётом адаптивности, что обеспечивает корректное отображение на различных устройствах. Для ускорения разработки применяется библиотека Element Plus, маршрутизация реализуется через Vue Router, а глобальное состояние пользователя, выбранных автомобилей и уведомлений поддерживается посредством Vuex.

Реализация серверной части выполняется на Node.js с помощью фреймворка NestJS, который обеспечивает структурированность и модульность кода. Разрабатываются контроллеры, сервисы и модули для обеспечения функциональности бронирования, аутентификации, управления тарифами и уведомлениями. Использование TypeORM обеспечивает удобную связь с MySQL, а применение JWT и ролевой модели доступа обеспечивает безопасную аутентификацию. Реализуется механизм логирования и обработки ошибок, а также интеграция со сторонними сервисами, включая платёжные шлюзы и системы уведомлений. Для обеспечения полноты документации API применяется OpenAPI.

На завершающем этапе проводится интеграция модулей и тестирование. Проверяется корректность взаимодействия frontend и backend через REST-интерфейсы. Тестирование включает модульные, интеграционные, системные и нагрузочные испытания, а также UX-оценку интерфейса. После устранения выявленных ошибок система разворачивается на серверной инфраструктуре – облачной или локальной. Настраиваются механизмы резервного копирования, мониторинга и технического сопровождения. Периодическая поддержка предполагает устранение ошибок, обновление функционала и обеспечение безопасности.



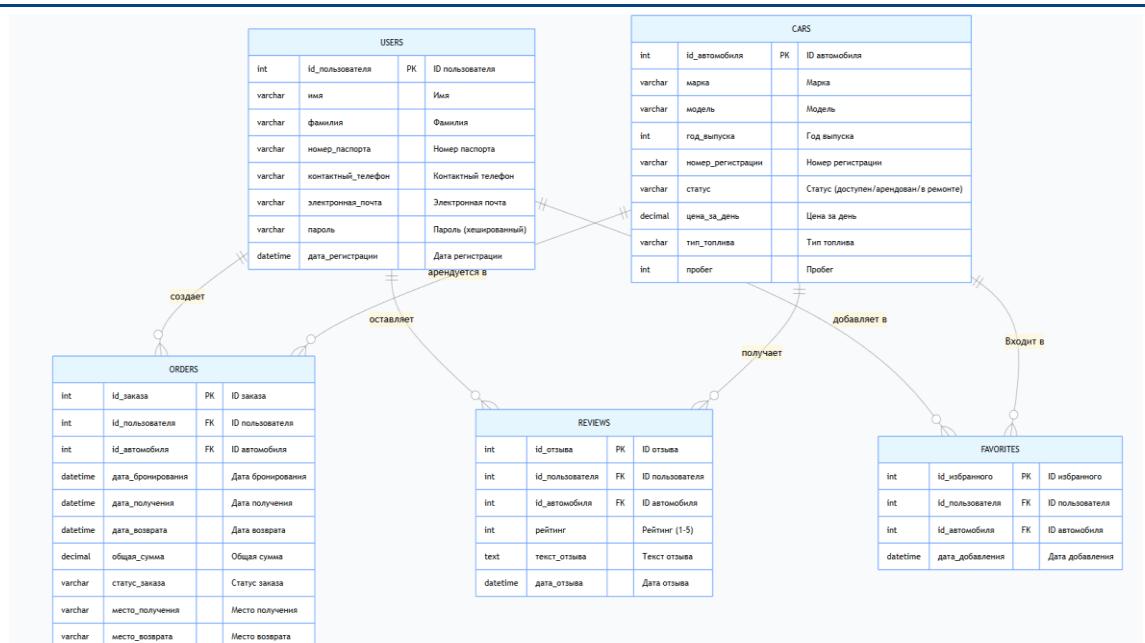


Рисунок 2. Логическая структура базы данных системы

Отдельное внимание в работе уделено анализу экономической целесообразности выбранного стека технологий. Node.js обеспечивает высокую пропускную способность при низком потреблении ресурсов, позволяя обрабатывать тысячи одновременных запросов на минимальной конфигурации сервера. Это снижает инфраструктурные затраты в сравнении с традиционными решениями. Клиентская отрисовка Vue.js уменьшает нагрузку на сервер, а гибкость развертывания на облачных сервисах минимизирует эксплуатационные расходы. Горизонтальная масштабируемость Node.js и вертикальная масштабируемость MySQL позволяют адаптировать систему к растущим требованиям без необходимости полной переработки архитектуры. Интеграция CI/CD сокращает сроки выпуска обновлений и уменьшает затраты на ручную работу.

Завершая исследование, следует отметить, что разработанная система проката автомобилей соответствует современным требованиям к веб-приложениям. Комплексность подхода – от анализа требований до развертывания – обеспечивает функциональность, устойчивость, высокую производительность и удобство использования. Технологический стек Vue.js, Node.js и MySQL подтверждает свою экономическую эффективность, обеспечивая гибкость, надёжность и возможность дальнейшего расширения. Система способна удовлетворить потребности пользователей и администраторов, обеспечивая удобные механизмы бронирования, управления автопарком и обработки заказов, а также устойчиво функционировать при увеличении нагрузки и расширении инфраструктуры.

Список литературы:

- «Онлайн-система проката автомобилей» – Miss Mansi Sanjay Jadhav, Mrunmesha Mangesh Karande.
- «Система бронирования проката автомобилей CarBy с микросервисной архитектурой» – Taha Abderrahmane Semmahi.
- «Онлайн система проката автомобилей с применением веб- и SMS-технологий» – Mohd Nizam Osman и др.
- «Построение децентрализованной системы аренды транспортных средств на базе Ethereum» – N. García-Moreno, P. Caballero-Gil, C. Caballero-Gil & J. Molina-Gil.

