

Организация интерактивного взаимодействия с объектами топливно-сервисной инфраструктуры с помощью мессенджеров

Карабанов А.Ф. (ИТМО)

Научный руководитель – Цопа Е.А. (ИТМО)

Введение

Современный рынок услуг автозаправочных станций (АЗС) и автомоек характеризуется наличием цифрового разрыва между традиционным офлайн-обслуживанием и существующими электронными сервисами. Традиционный способ сопряжен с временными затратами на ожидание в очередях и непосредственное взаимодействие с оператором. Существующие цифровые решения, в свою очередь, обладают существенным недостатком - они привязаны к конкретным платежным платформам и зачастую ограничены одним видом услуг, что вынуждает пользователя устанавливать и осваивать несколько различных приложений. Это приводит к фрагментации пользовательского опыта и нивелирует потенциальный выигрыш во времени. Целью данной работы является разработка подхода к организации взаимодействия с объектами топливно-сервисной инфраструктуры на базе мини-приложений в мессенджерах, обеспечивающего унификацию процесса и сокращение временных затрат пользователя.

Основная часть

В основе предлагаемого решения лежит гипотеза о том, что интеграция сервиса непосредственно в среду мессенджеров (Telegram, MAX) позволит снизить порог входа и ускорить обслуживание за счет использования уже существующей идентификации пользователя (Telegram, MAX ID) и привычного интерфейса. Для проверки гипотезы было разработано собственное мини-приложение.

Архитектура системы построена по принципам модульного монолита с элементами гексагональной архитектуры, что обеспечивает изоляцию доменной логики и простоту тестирования. Ключевым функциональным отличием от аналогов стала реализация двух независимых, но интегрированных в одном интерфейсе модулей: для заказа услуг АЗС и автомоек, а также поддержка оплаты через различные банки с использованием Системы быстрых платежей.

Проведенное сравнительное тестирование разработанного решения с существующими цифровыми аналогами показало его конкурентоспособность. Количественная оценка эффективности проводилась путем замера полного времени выполнения заказа - от момента открытия приложения до подтверждения оплаты. Традиционный способ обслуживания без использования приложений занимает в среднем 3 минуты. Разработанное приложение позволяет выполнить заказ в среднем за 1 минуту 30 секунд. При этом пользователю требуется совершить от 6 до 9 шагов в интерфейсе. Время создания заказа в разработанном приложении составляет 22 секунды, а время оплаты варьируется от 38 до 80 секунд в зависимости от выбранного платежного метода.

Дополнительным преимуществом разработанного решения является отсутствие необходимости в отдельной регистрации, благодаря использованию мессенджера как платформы идентификации. Проведенное нагрузочное

тестирование подтвердило способность системы выдерживать до 400 запросов в секунду, что обеспечивает стабильную работу при пиковых нагрузках.

Выводы

Результаты тестирования подтвердили эффективность предложенного подхода. Разработанное мини-приложение демонстрирует сопоставимое с ведущими рыночными решениями время обслуживания, составляющее в среднем 1 минуту 30 секунд, при этом выгодно отличается от них функционально, предлагая пользователю бесшовный доступ как к услугам АЗС, так и автомоек, а также свободу выбора платежного инструмента. Использование мессенджера в качестве платформы позволило сократить путь пользователя до 6–9 шагов, что особенно заметно при использовании оплаты через Систему быстрых платежей с привязанным счетом. Дальнейшее развитие работы будет направлено на расширение географии сервиса и интеграцию с программами лояльности партнеров для улучшения показателя общего времени обслуживания.

Литература

1. Вирух Р. Путь к React. – 5-е издание. – Самиздат, 2022. – 250 с.
2. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. – СПб.: Питер, 2020. – 544 с.
3. Эванс Б., Кларк Д., Фербург М. Java для опытных разработчиков. – 2-е изд. – Санкт-Петербург: Питер, 2024. – 736
4. PostgreSQL Documentation [Электронный ресурс] // The PostgreSQL Global Development Group. – URL: <https://www.postgresql.org/docs/> (дата обращения: 24.02.2026).
5. Prometheus. Monitoring system and time series database. Documentation [Электронный ресурс] // Prometheus. – URL: <https://prometheus.io/docs/> (дата обращения: 24.02.2026).
6. Grafana documentation [Электронный ресурс] // Grafana Labs. – URL: <https://grafana.com/docs/grafana/latest/> (дата обращения: 24.02.2026).
7. Docker documentation [Электронный ресурс] // Docker. – URL: <https://docs.docker.com/> (дата обращения: 24.02.2026).
8. Spring Boot Reference Documentation [Электронный ресурс] // VMware. – URL: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> (дата обращения: 24.02.2026).
9. 12 OWASP Top Ten: The Ten Most Critical Web Application Security Risks [Электронный ресурс] // OWASP Foundation. – URL: <https://owasp.org/www-project-top-ten/> (дата обращения: 24.02.2026).
10. Telegram Bot API [Электронный ресурс] // Telegram. – URL: <https://core.telegram.org/bots/api> (дата обращения: 24.02.2026).