

Оглавление

1	Введение.....	2
2	Итоги 2 аттестации.....	2
2.1	Разработка MVP.....	2
2.1.1	Модуль «Авторизация/аутентификация»	2
2.1.2	Модуль «Компания».....	3
2.1.3	Модуль «Автоответчик»	3
2.1.4	Модуль «Подписки на сервис»	3
2.1.5	Модуль «AI ассистента»	4
2.1.6	Модуль «Интеграции с внешними API»	4
2.1.7	Модуль «Диалоги»	4
2.1.8	Связь модулей и взаимодействие	5
2.2	Разработка Front-end.....	5
2.3	CI/CD-процесс.....	7
2.4	Тестовая документация	7
2.5	Проблемы и решения	7
2.6	Итоги этапа.....	8
3	Планы на третий этап (Этап предзащиты)	9

1 Введение

Проект «Автоответчик для почты и других чатов» (DialogX), который представляет собой веб-сайт приложение для автоматизации коммуникации, продолжает свое развития. По состоянию на текущий момент завершается этап второй аттестации. В данном отчете представлены:

- результаты выполнения работ на втором этапе;
- анализ возникших сложностей и нахождение их решений;
- план проекта на третью аттестацию;
- выделение критериев успешности 3 аттестации.

2 Итоги 2 аттестации

2.1 Разработка MVP

На этапе второй аттестации основной задачей было создание MVP приложения. На данный момент реализован MVP продукт, в котором присутствует основные пользовательские сценарии. Пользователь может пройти регистрацию, авторизацию, подключить интеграции с мессенджерами, управлять шаблонными ответами, использовать чаты, оформить подписку.

2.1.1 Модуль «Авторизация/аутентификация»

В рамках данного модуля, был реализован JWT-флоу: регистрация, авторизация пользователей, а также обновление токена, которое позволяет обеспечить безопасную работу сессий.

Для повышения безопасности и удобства управления сессиями была реализована система хранения токенов в Redis.

Система поддерживает три основных роли: пользователь (User), оператор (Operator), администратор (Admin).

2.1.2 Модуль «Компания»

Модуль "Компания" отвечает за управление операторами и настройку приложения. Основная задача данного компонента — интерфейс для настройки работы сервиса и управления операторами.

На втором этапе были реализованы основные функции: создание компании, редактирование ее параметров, добавление операторов, удаление операторов, данный модуль пока что имеет только backend реализацию.

2.1.3 Модуль «Автоответчик»

Был реализован модуль автоответчика. Модуль автоответчика представляет собой связующее звено между AI-ассистентом и шаблонной системой ответов. Его задача — обеспечить осмысленную реакцию на входящее сообщение пользователя даже в случае недоступности AI.

Работа модуля строится по следующей логике: при поступлении сообщения сначала выполняется попытка получить AI-ответ; если это невозможно, система автоматически выдает наиболее подходящий шаблонный ответ.

2.1.4 Модуль «Подписки на сервис»

Был реализован модуль подписки на сервис. Система подписок является ключевым элементом монетизации проекта и обеспечивает пользователю доступ к различной функциональности.

В рамках второго этапа была реализована базовая логика работы подписок с тарифами «Тестовый», «Соло» и «Команда». Пользователь может просматривать доступные тарифы, активировать их.

Функция оплаты была эмулирована через заглушку, имитирующую оплату.

2.1.5 Модуль «AI ассистента»

Был реализован один из ключевых модулей, предоставляющий интеллектуальные ответы в чате. Он взаимодействует с внешним AI API, обрабатывая сообщения пользователей и формируя осмысленные ответы, на основе контекста шаблонного ответа.

AI-ассистент также интегрирован с модулем автоответчика: если система получает отрицательный результат от AI, она обращается к шаблонной базе, чтобы предоставить пользователю наиболее подходящий шаблонный ответ.

2.1.6 Модуль «Интеграции с внешними API»

Был разработан модуль интеграций. Его задача — обеспечить связку между нашим приложением и внешними каналами связи, такими как мессенджеры, email-сервисы. Это позволяет компаниям обрабатывать входящие сообщения от клиентов через привычные каналы, оставаясь при этом в рамках единого интерфейса.

В рамках второго этапа аттестации были реализованы следующие интеграции:

- email (@yandex.ru и @gmail.ru) - пользователи получили возможность подключать свои почтовые аккаунты для приема и отправки сообщений, переписка сохраняется в базе данных;
- telegram- подключение осуществляется через токен бота. Пользователь указывает токен, после чего система настраивает webhook и начинает принимать сообщения, отправка осуществляется через Telegram Bot API.

2.1.7 Модуль «Диалоги»

Был реализован модуль диалогов. Он реализует ключевую функциональность: обмен сообщениями, отображение истории чатов,

реакцию на новые сообщения и поддержку многоканальной переписки. Это основной рабочий инструмент как для операторов, так и для администраторов.

Каждое сообщение, отправленное пользователем или клиентом, сохраняется в базе данных PostgreSQL.

А также реализована поддержка WebSocket.

2.1.8 Связь модулей и взаимодействие

В архитектуре основной акцент сделан на разделение слоев — фронтенда, серверной логики (backend) и взаимодействия с внешними API. Это позволяет добиться, масштабируемости проекта.

Взаимодействие между frontend и backend осуществляется через REST API. Каждый модуль предоставляет набор REST-эндпоинтов, которые обрабатываются контроллерами и передаются в соответствующие сервисы.

Для модуля «Диалоги» было реализовано WebSocket-соединение, что позволило организовать real-time обмен сообщениями. Это особенно важно для операторов, работающих с активными чатами.

Коммуникация между внутренними модулями backend осуществляется через сервисный слой. Каждый модуль инкапсулирует бизнес-логику, работает с репозиториями и взаимодействует с другими модулями только через явно описанные интерфейсы. Такой подход исключает прямые зависимости между модулями и способствует их переиспользованию и тестируемости.

2.2 Разработка Front-end

Разработка клиентской части велась с учетом макетов Figma, а также требований брендбука и UI Kit.

Были разработаны наиболее приоритетные страницы авторизации и регистрации, диалогов, главная страница, страница интеграций, страница шаблонных ответов, приветственная страница.

Основным фреймворком для реализации был выбран React.

При открытии сайта пользователь попадает на информационную страницу-приветствие, где ему предоставляется краткий обзор возможностей сервиса, преимуществ использования, а также доступ к регистрационной и авторизационной форме.

После регистрации система автоматически перенаправляет пользователя на страницу авторизации.

Страница входа включает механизм восстановления доступа по email и обратную навигацию на регистрацию.

После успешной авторизации пользователь попадает в рабочую область приложения — главную страницу. Главная страница разработана в виде следующего интерфейса: слева — список диалогов, по центру — активное окно переписки, а также панель быстрого доступа к ключевым модулям.

Страница интеграций позволяет подключить интеграции и визуализирует подключенные каналы связи — на данный момент реализованы интеграции с Telegram, Email (@gmail.ru, @yandex.ru). VK, WhatsApp и виджет будут реализованы на третьем этапе. Подключение каждого канала осуществляется через отдельную форму.

На странице подписки пользователь может просматривать информацию о тарифах, а также оформить подписку.

Модуль шаблонов позволяет администратору загружать шаблонные ответы или редактировать их вручную на сайте.

На странице «Диалоги» отображается активный список чатов и текущее общение с пользователем.

Раздел «Статистика» будет реализован в будущем.

Макеты страниц были разработаны в Figma. На их основе была произведена реализация интерфейсов на React.

Интеграция с backend осуществлялась поэтапно, начиная с наиболее значимых компонентов.

2.3 CI/CD-процесс

Для обеспечения стабильности и скорости релизов в нашем проекте реализован минимальный, но функциональный процесс CI/CD на базе GitHub Actions.

Таким образом, весь процесс от коммита до деплоя минимизирует участие команды, позволяя оперативно вносить изменения и снижать количество ошибок на продакшене.

2.4 Тестовая документация

В рамках разработки проекта была подготовлена базовая тестовая документация, отражающая текущее состояние проверок и стабильности системы. Основное внимание было уделено покрытию бизнес-логики через юнит- и интеграционные тесты.

Для backend-модулей написан набор юнит-тестов, покрывающий ключевые компоненты.

Кроме того, проведено ручное функциональное тестирование основных пользовательских сценариев через Postman. Проверялась регистрация, авторизация, отправка сообщений, подключение интеграций и работа автоответчика.

2.5 Проблемы и решения

В процессе разработки проекта возникло несколько организационных и технических затруднений, которые повлияли на ход реализации.

Одной из ключевых проблем стало отсутствие возможности найти подходящий бесплатный сервер для размещения backend-части приложения. Попытки использовать бесплатные или trial-хостинги оказались безуспешными: ресурсы были ограничены по оперативной памяти, не поддерживали Docker или не обеспечивали стабильного интернет-соединения.

В итоге было принято решение приобрести VPS-сервер с необходимыми характеристиками. Это позволило развернуть рабочее окружение, настроить CI/CD и обеспечить постоянную доступность приложения, но повлекло за собой дополнительные затраты.

Также в процессе интеграции Telegram возникли трудности. API Telegram несмотря на общую популярность и документацию, периодически демонстрировал нестабильность. Проблема была решена путём настройки механизма повторной отправки и увеличения таймаутов ожидания.

Дополнительные затруднения возникли при реализации платежа в подписке — отсутствие доступа к полноценному тестовому стенду платёжной системы ограничивало возможности полной отладки. Была реализована заглушка оплаты.

2.6 Итоги этапа

На момент завершения второго этапа команда завершила интеграцию основных модулей проекта и достигла работы системы в рамках MVP. Ключевые функциональные компоненты, включая авторизацию, регистрацию, работу с диалогами, AI-ассистентом, автоответчиком, а также модуль интеграций и подписок, были реализованы.

Были проведены функциональные и модульные тестирования, позволившие выявить и устранить критические ошибки. Разработка фронтенда позволила сформировать пользовательский интерфейс, обеспечивающий доступ к функциям. Приложение успешно развернуто на удалённом VPS-сервере с использованием Docker-контейнеров и автоматизированным CI/CD-процессом

Работа с внешними API позволила наладить интеграцию с Telegram, Email. Несмотря на ряд технических и организационных сложностей (включая покупку сервера и нестабильность Telegram API), задачи этапа были выполнены.

Проект готов к демонстрации MVP версии. И переходу на следующую фазу предзащиты.

3 Планы на третий этап (Этап предзащиты)

На третьем этапе проекта мы ставим перед собой задачу подготовить полнофункциональную версию системы DialogX, доступную для демонстрации и внешнего тестирования. Приложение должно включать все ключевые модули, работать стабильно, иметь завершённый интерфейс, документацию и первичную систему аналитики.

С точки зрения клиентской части будет полностью реализован интерфейс в соответствии с макетами в Figma и рекомендациями UI Kit. Мы также проведем валидацию форм, реализуем уведомления об ошибках и доведем до конца интеграцию с backend-методами.

На серверной стороне предстоит провести доработку бизнес-логики. API будут задокументированы в Swagger, чтобы упростить тестирование и интеграцию. Мы также планируем оптимизировать работу с базой данных и кешем, а также обеспечить интеграцию с оставшимися мессенджерами.

Планируется расширить существующие юнит- и интеграционные тесты. Результаты будут зафиксированы в отчете, а найденные ошибки - исправлены.

В рамках третьего этапа также планируется подключение систем аналитики. Мы интегрируем метрику, предварительно это будут Grafana и Prometheus и настроим сбор ключевых событий. На основе собранных данных сформируем предварительный отчет, который поможет понять, как пользователи взаимодействуют с системой.

Таким образом, третий этап станет завершающим перед публичной демонстрацией проекта, и наша цель — довести модули до максимально готового и стабильного состояния.

Отчет подготовлен: Баранником Данилом Евгеньевичем

Дата: 05.05.2025

