ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ МЕНЕДЖЕРОВ ОТДЕЛА ПРОДАЖ – AI МОП

г. Москва

2025

ОГЛАВЛЕНИЕ

1 ОБЩИЕ СВЕДЕНИЯ

Полное наименование: Программное обеспечение для проектирования интеллектуальных менеджеров отдела продаж – AI МОП (далее Система).

1.1 Цели создания системы

Автоматизация процессов продаж для повышения эффективности отдела продаж, уменьшения издержек и роста конверсии за счёт внедрения ИИ-менеджеров, способных заменить или дополнить работу живых сотрудников.

1.2 Назначение системы

Выполнение функций менеджера по продажам с использованием искусственного интеллекта, обеспечивая эффективное взаимодействие с клиентами на всех этапах воронки продаж.

2 АРХИТЕКТУРА ПРОЕКТА

Система разработана в рамках микросервисной архитектуры с использованием современных технологий и фреймворков для обеспечения эффективности, масштабируемости и удобства в поддержке. Архитектура состоит из следующих ключевых компонентов:

1. Backend (FastAPI):

– Фреймворк FastAPI используется для разработки серверной части приложения, обеспечивает удобное создание API для взаимодействия с Frontend-частью Системы.

2. База данных (PostgreSQL):

– PostgreSQL выбрана в качестве системы управления базами данных для обеспечения высокой надежности и поддержки сложных запросов.

– SQL Alchemy ORM используется для удобного взаимодействия с базой данных, создания моделей данных и выполнения запросов.

– Внедрение миграций Liquibase обеспечивает контроль версий схемы базы данных и упрощает процесс обновления структуры данных.

3. Web Frontend (React):

– Frontend разработан с использованием фреймворка React для создания динамичного пользовательского интерфейса.

– Используются компоненты Material UI для создания модульной структуры приложения.

– Для управления состоянием приложения используется Redux.

– Маршрутизация реализована с использованием React Router для навигации по экранам.

– Интерактивные элементы интерфейса обеспечивают приятный пользовательский опыт.

4. Контейнеризация:

– Применение Docker-контейнеров для сервисов Системы обеспечивает легкость развертывания, масштабирования и управления зависимостями приложения.

– Openshift используется для управления многоконтейнерными приложениями.

Эта архитектура предоставляет гибкую и масштабируемую основу для развертывания, обеспечивая эффективное взаимодействие между всеми сервисами проекта.

3 ПЕРЕЧЕНЬ ПОДСИСТЕМ, ИХ НАЗНАЧЕНИЕ И ОСНОВНЫЕ ХАРАКТЕРИСТИКИ

3.1 Подсистема авторизации пользователей

Подсистема авторизации пользователей представляет собой компонент Системы, который отвечает за управление доступом пользователей к различным ресурсам и функциям. Она обеспечивает проверку тарифа пользователя на выполнение определённых действий и доступ к функциям. Основные функции подсистемы авторизации включают:

1) Аутентификация;
2) Авторизация;
3) Контроль и управление сессиями;
4) Логирование и мониторинг.

3.2 Подсистема навигации пользователя в боковом меню

Подсистема навигации пользователя в боковом меню позволяет пользователю перемещаться по разделам Системы и представляет собой набор вкладок:

1) дашборд;
2) обучение AI агентов;
3) рабочий интерфейс;
4) аналитика;
5) события;
6) интеграции.

3.3 Подсистема обучения AI агентов

Обеспечивает обучение и самообучение интеллектуальных агентов на основе историй продаж, диалогов и пользовательских сценариев. Позволяет настраивать поведение агентов, формировать базы знаний и улучшать качество коммуникаций за счёт машинного обучения и анализа успешных кейсов.

3.4 Подсистема сценариев (работы AI менеджеров отдела продаж)

Отвечает за создание, хранение и выполнение диалоговых сценариев, по которым действуют AI-менеджеры. Позволяет моделировать последовательность шагов общения с клиентом, учитывать ответы, намерения и контекст, обеспечивая естественную логику разговора и движение клиента по воронке продаж.

3.5 Подсистема аналитики

Собирает и анализирует данные о работе агентов, результатах диалогов, конверсии и эффективности сценариев. Формирует отчёты и визуализации, помогает оценивать качество коммуникаций, выявлять слабые места и оптимизировать стратегию продаж.

3.6 Подсистема событий

Отслеживает и регистрирует все действия и события, происходящие в системе (начало/окончание диалога, изменение стадии сделки, входящие сообщения и т. д.). Позволяет инициировать автоматические реакции или триггеры – например, уведомления, обновление данных в CRM или запуск нового сценария.

3.7 Подсистема интеграций

Обеспечивает взаимодействие AI МОП с внешними системами — CRM, телефонией, мессенджерами, почтой и аналитическими сервисами. Позволяет обмениваться данными, синхронизировать статусы клиентов и автоматизировать действия в сторонних приложениях.

3.8 Функциональные требования к программному обеспечению

Подсистема обучения AI-агентов:

Назначение: обеспечение обучения и самообучения интеллектуальных агентов на основе данных продаж и диалогов.

Требования:

1. Система должна обеспечивать загрузку и хранение обучающих данных (тексты диалогов, примеры успешных продаж, FAQ, клиентские профили).
2. Система должна поддерживать настройку параметров обучения (тональность, стиль общения, цели коммуникации).
3. Система должна обеспечивать автоматическое самообучение агентов на основании результатов их взаимодействий с клиентами.
4. Система должна предоставлять интерфейс для оценки качества обучения и сравнения версий моделей.
5. Должна быть реализована возможность ручного дообучения агента пользователем (корректировка ответов, добавление новых примеров).

Подсистема сценариев (работы AI-менеджеров отдела продаж):

Назначение: управление диалоговыми сценариями, определяющими поведение агентов при взаимодействии с клиентами.

Требования:

1. Система должна позволять создавать, редактировать и сохранять сценарии диалогов в визуальном и текстовом формате.
2. Система должна поддерживать ветвление логики сценариев в зависимости от ответов клиента, стадии сделки и контекста.
3. Сценарии должны обеспечивать выполнение автоматических действий (например, создание задачи в CRM, отправка сообщения, переключение стадии сделки).
4. Система должна поддерживать мультиязычные сценарии.
5. Должна быть предусмотрена возможность тестирования и симуляции диалогов до их запуска в рабочую среду.

Подсистема аналитики:

Назначение: сбор, обработка и визуализация статистических данных о работе AI-агентов и эффективности продаж.

Требования:

1. Система должна собирать данные о количестве обработанных лидов, продолжительности диалогов, конверсии, результатах сделок.
2. Система должна формировать отчёты по выбранным периодам, сценариям и агентам.
3. Система должна предоставлять панель мониторинга (dashboard) с основными метриками в реальном времени.
4. Должна быть реализована возможность экспорта отчётов в форматы CSV, XLSX, PDF.
5. Система должна поддерживать визуализацию данных (графики, диаграммы, таблицы).

Подсистема событий:

Назначение: фиксация, хранение и обработка событий, происходящих в системе и во взаимодействиях с внешними сервисами.

Требования:

1. Система должна регистрировать все события (начало/окончание диалога, смена стадии сделки, поступление сообщения, ошибка).
2. Система должна обеспечивать фильтрацию и поиск событий по типу, дате и участнику.

3. Система должна обеспечивать настройку реакций на события (например, уведомления, вызов API, запуск сценария).
4. Система должна хранить историю событий не менее чем за 12 месяцев.
5. Должна быть реализована возможность экспорта журнала событий.

Подсистема интеграций

Назначение: обеспечение взаимодействия с внешними системами и источниками данных.

Требования:

1. Система должна поддерживать интеграции с популярными CRM (Bitrix24, amoCRM).
2. Система должна обеспечивать обмен данными с мессенджерами (Telegram, WhatsApp) и телефонией.
3. Должна быть реализована возможность интеграции через REST API и Webhooks.
4. Система должна обеспечивать синхронизацию статусов клиентов и сделок в обе стороны.
5. Должна поддерживаться авторизация и безопасная передача данных по протоколу HTTPS.

3.9 Нефункциональные требования к программному обеспечению

Требования к производительности:

1. Система должна обеспечивать одновременную работу не менее 10 000 активных диалогов без снижения качества обслуживания.
2. Время отклика агента на сообщение клиента не должно превышать 3 секунд.
3. Система должна поддерживать обновление данных в CRM в течение не более 5 секунд после события.

Требования к надёжности и отказоустойчивости:

1. Система должна обеспечивать сохранность данных при сбоях и перезапуске.
2. Время недоступности системы не должно превышать 0,1% в месяц.
3. Должна быть реализована система резервного копирования и восстановления данных.

Требования к безопасности:

1. Все данные пользователей и клиентов должны передаваться по защищённым каналам (HTTPS, SSL/TLS).

2. Система должна поддерживать ролевую модель доступа (администратор, аналитик, оператор).
3. Должен вестись аудит действий пользователей.
4. Пароли и токены должны храниться в зашифрованном виде.

Требования к совместимости и масштабируемости:

1. Система должна быть совместима с основными веб-браузерами и мобильными устройствами.
2. Архитектура должна позволять горизонтальное масштабирование (добавление серверов и агентов).
3. Возможность интеграции с внешними системами без существенной доработки ядра платформы.

Требования к удобству использования (UX/UI):

1. Интерфейс должен быть интуитивно понятным и адаптивным под различные устройства.
2. Должна быть реализована контекстная справка и документация.
3. Визуальные панели (дашборды, сценарии, аналитика) должны быть доступны без необходимости программирования.

Требования к сопровождению и обновлению:

1. Система должна поддерживать удалённое обновление без остановки сервиса.
2. Обновления не должны нарушать сохранённые данные и настройки.
3. Должна быть реализована система логирования и диагностики ошибок.

4 ТРЕБОВАНИЯ К СЕРВЕРНОЙ ИНФРАСТРУКТУРЕ

Контейнеризация приложений в OpenShift требует соблюдения ряда принципов и стандартов, чтобы гарантировать эффективную работу и управление контейнерами. Вот основные требования:

1. Изоляция приложений: контейнеры должны быть изолированы друг от друга и от хост-системы.

2. Использование образов контейнеров: приложения должны быть упакованы в образы контейнеров, которые могут быть созданы с использованием Dockerfile или других инструментов сборки. Эти образы должны быть доступны в реестре контейнеров.

3. Поддержка многопоточности и масштабируемости: приложения должны быть спроектированы для работы в распределенной среде, что включает возможность горизонтального масштабирования.

4. Сетевые настройки: приложения должны использовать сетевые политики OpenShift для управления сетевым трафиком и обеспечения безопасности.

5. Хранение данных: для хранения данных необходимо использовать постоянные тома (Persistent Volumes) и соответствующие классы хранения (Storage Classes).

6. Мониторинг и логирование: приложения должны быть настроены для интеграции с системами мониторинга и логирования, такими как Prometheus.

7. Безопасность: следует учитывать безопасность на всех уровнях, включая использование Security Contexts, ограничение прав доступа и применение политик безопасности.

8. Конфигурация и секреты: конфигурационные данные и секреты должны храниться в ConfigMaps и Secrets соответственно, чтобы обеспечить безопасное управление конфиденциальной информацией.

9. CI/CD интеграция: рекомендуется интегрировать приложения с CI/CD пайплайнами для автоматизации сборки, тестирования и развертывания.

10. Соблюдение стандартов и практик DevOps: важно следовать лучшим практикам DevOps при разработке, развертывании и управлении приложениями.

Соблюдение этих требований поможет обеспечить стабильную и безопасную работу приложений в среде OpenShift.

5 ТРЕБОВАНИЯ К НАДЕЖНОСТИ И БЕЗОПАСНОСТИ ТЕХНИЧЕСКИХ СРЕДСТВ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Требования к надежности технических средств и программного обеспечения включают в себя несколько аспектов, целью которых является обеспечение стабильности и бесперебойной работы системы, а именно:

1. Доступность: системы должны быть доступны пользователям в любое время, согласно установленным уровням обслуживания (SLA). Это включает в себя минимизацию времени простоя.

2. Отказоустойчивость: системы должны быть спроектированы так, чтобы продолжать функционировать даже в случае сбоев отдельных компонентов. Это может включать резервирование, дублирование и автоматическое переключение на резервные системы.

3. Устойчивость к нагрузке: программное обеспечение должно быть способно обрабатывать ожидаемую и даже повышенную нагрузку без снижения производительности.

4. Поддержка и обслуживание: надежные системы должны иметь возможность для легкого обновления, исправления ошибок и масштабирования без значительных перебоев в работе.

5. Тестирование и валидация: тестирование (включая нагрузочные и стресс-тесты) должно проводиться для выявления потенциальных проблем до их появления в рабочей среде.

Требования к безопасности включают в себя следующие основные пункты:

1. Аутентификация и авторизация: необходимо обеспечить надежные механизмы аутентификации пользователей и контроля доступа к системам и данным.

2. Шифрование данных: данные должны быть защищены как в состоянии покоя, так и при передаче. Использование протоколов шифрования TLS является обязательным.

3. Защита от угроз: системы должны иметь средства защиты от различных угроз, таких как вирусы, вредоносное ПО, атаки DDoS и другие виды кибератак.

4. Мониторинг и аудит: необходимо внедрить системы мониторинга для отслеживания подозрительной активности и ведения журналов для последующего анализа.

5. Обновления и патчи: регулярное применение обновлений безопасности и исправлений для программного обеспечения и операционных систем.

6. Обучение пользователей: пользователи должны быть обучены основам безопасности, включая распознавание фишинга и безопасное использование систем.

7. Политики безопасности: разработка и внедрение документированных политик безопасности, которые определяют правила и процедуры для защиты информации.

8. Резервное копирование данных: регулярное создание резервных копий критически важных данных для обеспечения их восстановления в случае потери или повреждения.

Соблюдение этих требований обеспечит как надежность, так и безопасность технических средств и программного обеспечения, что, в свою очередь, повысит доверие пользователей и снизит риски для бизнеса.