

Введение

- 1. Цель документа:** Установить и зафиксировать четкие однозначные требования к разрабатываемому продукту, установить взаимопонимание внутри команды во время разработки продукта и избежать двусмыслинности. Документ предназначен для использования в качестве основы на всех этапах жизненного цикла продукта.
- 2. Область применения системы:** Продукт будет доступен для использования в виде клиент-серверного веб-приложения, которое поможет пользователю осваивать команды для bash-терминала в GNU/Linux.
- 3. Ссылки на другие документы/стандарты:** ISO/IEC/IEEE 29148:2018.

Общая характеристика системы

1. Перспектива системы. Разрабатываемая система является клиент-серверным веб-приложением с чат-ботом на основе LLM-модели, предназначенным для помощи в освоении bash-терминала в GNU/Linux. Клиентская часть работает в браузерах на ПК. Серверная часть развернута на сервере и обрабатывает поступающие запросы из клиентской части.

2. Функции системы. Продукт обладает следующими функциями:

- Генерирование команды/набора команд для bash-терминала в GNU/Linux по запросу пользователя.
- Разбор команды/набор команд для bash-терминала в GNU/Linux по запросу пользователя.
- Регистрация/авторизация пользователя.

3. Пользователи и заинтересованные стороны.

- Домашние пользователи дистрибутивов Linux.
- Начинающие пользователи дистрибутивов Linux.
- Разработчики и инженеры, работающие в дистрибутивах Linux.
- Системные администраторы.

4. Технологический стек.

- **Frontend:** React, Tailwind CSS, JavaScript, REST API.
- **Backend:** Golang, Redis, REST API/WebSocket.
- **ML-сервис:** Python, PyTorch, Transformers, Hydra, FastAPI, WebSocket.
- **База данных:** PostgreSQL.
- **Инфраструктура:** Docker, Docker Compose, Nginx.

5. Предположения и зависимости.

Предполагается, что:

- Пользователь знаком с Linux-дистрибутивами.
- Пользователь пользуется веб-приложением с десктопного браузера.
- Веб-приложение развернуто локально на ПК для демонстрации и на сервере для продакшена.

- ML-модель предобучена, и требуется её дообучить.

Ограничения:

- Чат-бот работает только с командами в GNU/Linux.
- Модель сжата до 8-bit для инференса и обучения.
- Ответ чат-бота не гарантирует 100% правильности и безопасности.

Функциональные требования

- 1.** Регистрация с помощью электронной почты.
- 2.** Обращение к чат-боту и получение ответ от него.
- 3.** Новая генерация ответа чат-ботом, вместо старого.
- 4.** При получении чат-ботом команды/набора команд:
 - Подробный рассказ, что делает каждая команда и за что отвечает каждый флаг и параметр в полученных командах.
 - Если команда/набор команд, содержит ошибки, то чат-бот указывает на них, исправляет, пишет рабочий вариант, а дальше выполняет, первый пункт.
 - Если команда/набор команд являются небезопасными, не актуальными или имеются альтернативы лучше, то чат-бот выполняет первый пункт и второй пункт (при необходимости), после чего он предупреждает пользователя о проблеме команды/набора команд и предлагает альтернативу, если такое возможно. Для альтернативы также требуется выполнить первый пункт.
 - Если пользователь ввел запрос на помощь с составлением команд, то чат-бот составляет команду/набор команд, решающий проблему пользователя, выполнив первый пункт.
 - Если пользователь ввел запрос, не связанный с помощью с bash-терминалом в GNU/Linux, то чат-бот отвечает ему, что он не может помочь с его проблемой, так как специализируется только на помощи с bash-терминалом в GNU/Linux.
- 5.** Оценка ответа чат-бота (лайк/дизлайк), после чего, если ответ был оценен, он заносится в базу данных в таблицу message_feedbacks.
- 6.** Обращение к своим прошлым чатам.
- 7.** Возможность удаления старых чатов.
- 8.** Возможность изменить модель чат-бота (Qwen2.5-1.5B-Instruct, Qwen2.5-7.5B-Instruct)

Нефункциональные требования

1. Производительность:

- Продукт должен выдерживать работу 100 пользователей одновременно.
- Время отклика сервиса должно составлять 1-2 секунды.
- Пользователь должен получить ответ чат-бота в течение 7-10 секунд.

2. Надежность.

- Среднее время безотказной работы $\geq 99\%$ в месяц.

3. Безопасность.

- Хранение паролей пользователей в виде хэшов, полученных с помощью метода SCRAM-SHA-256.

4. Масштабируемость.

- Продукт должен иметь возможность добавления новых модулей и улучшения старых без переработки архитектуры.

5. Поддерживаемость.

- Возможность изменять каждый модуль не влияя на работу остальных.

6. Совместимость.

- Совместимость с браузерами Google Chrome, Firefox, Microsoft Edge, Yandex.

Интерфейсы

1. Пользовательский интерфейс.

Веб-приложение должно содержать 4 страницы: Главная страница с описанием продукта, страница с чат-ботом, страница авторизации и страница регистрации.

Страница авторизации:

- Отображение ошибки при вводе некорректной электронной почты во время авторизации.
- Отображение ошибки при вводе некорректного пароля во время авторизации.
- Отображение ошибки при вводе неизвестного юзернейма во время авторизации.
- Ввод пароля должен скрываться символом «*», но должна иметься возможность отключить это.
- При нажатии кнопки «Войти» после успешной аутентификации пользователь должен попасть на страницу с чат-ботом.
- Под кнопкой «Войти» располагается гиперссылка «Нет аккаунта? Зарегистрируйтесь», которая ведет пользователя на страницу регистрации.

Страница регистрации:

- Отображение сообщения о том, что код отправлен на почту при регистрации.
- Отображение ошибки при вводе некорректного кода при регистрации.
- Отображение ошибки при вводе существующего юзернейма при регистрации.
- Ввод пароля должен скрываться символом «*», но должна иметься возможность отключить это.
- При нажатии кнопки «Зарегистрироваться» пользователь должен попасть на страницу с чат-ботом.

- Под кнопкой «Зарегистрироваться» располагается гиперссылка «Уже есть аккаунт? Войдите», которая ведет пользователя на страницу авторизации.

Главная страница:

- Информация о том, какую проблему решает наше клиент-сервисное веб-приложение.
- Слева в шапке страницы располагается логотип, при нажатии на который пользователь попадает на главную страницу, либо, если он уже на ней, она обновляется.
- Справа в шапке располагаются кнопки «Регистрация», «Авторизация», «Главная страница».
- При нажатии кнопки «Регистрация» пользователь должен попасть на страницу регистрации.
- При нажатии кнопки «Авторизация» пользователь должен попасть на страницу авторизации.

Страница с чат-ботом:

- В центре страницы располагается многострочное поле для ввода запроса, которое представлено прямоугольником с закругленными краями. Поле расширяется по мере ввода запроса пользователем.
- Над полем для ввода запроса должна быть надпись «Чем сегодня я могу Вам помочь?». Над этой надписью располагается текст «TuxBot AI Assistant».
- При появлении текста в поле для ввода запроса должна появиться кнопка «Отправить».
- При нажатии на кнопку «Отправить» страница должна динамически обновиться и стать похожей на диалог в мессенджере, где с правой стороны страницы располагаются сообщения пользователя, а с левой стороны страницы располагаются ответы чат-бота. Запрос должен быть отправлен модели, сгенерированный ответ который должен быть отправлен обратно на клиентский сервер.

- При наведении на сообщение бота снизу него должны появится иконки лайка/дизлайка, при нажатии на которые сообщения заносятся в message_feedbacks. Также должна появится иконка зацикленной стрелки, при нажатии на которую генерируется другой ответ на тот же самый вопрос.
- После генерации первого ответа в новом чате он заносится в базу данных.
- С левой стороны страницы располагается вертикальное поле, внизу которого расположена кнопка выхода из аккаунта, а сверху расположена кнопка создания нового чата. Между ними расположены старые чаты.
- При наведении курсора на старый чат появляется иконка корзины, при нажатии на которую появляется предупреждающее сообщение: если выбрать «Да», то чат удаляется, если выбрать «Нет», то сообщение пропадает без каких-либо дополнительных действий.
- При нажатии кнопки выхода из аккаунта пользователь должен получить предупреждающее сообщение, в котором при нажатии кнопки «Да» пользователь попадает на главную страницу, а при нажатии кнопки «Нет» — предупреждающее сообщение пропадает без каких-либо дополнительных действий.
- При нажатии кнопки создания нового чата страницы должна динамически обновиться и появится поле для ввода запроса по середине экрана с текстом над ним, как описано в первом и втором пунктах.
- В верхнем правом углу пространства для чата расположен выпадающий список с доступными моделями, а также показывается текущая используемая модель (по умолчанию Qwen-2.5-1.5B-Instruct).

2. Программные интерфейсы (API).

Система должна предоставлять REST API для операций с пользователями, которое предназначено для использования через протокол HTTPS. Запросы и ответы должны быть в формате JSON.

HTTP-запросы:

- **POST /users** Входные данные: name, surname, email, password.
- **POST /users/verifications** Входные данные: code.
Выходные данные: access_token (срок действия 15 минут) и refresh_token (срок действия 1 день). Передаются в заголовке.
- **POST /auth/login**. Входные данные: email, username, password.
Выходные данные: access_token (срок действия 15 минут), refresh_token. Передаются в заголовке.
- **POST /auth/logout**. Входные данные: access_token, refresh_token (удаляется из cookie и из таблицы user_sessions). Передаются в заголовке.
- **POST /chats**. Входные данные: access_token (в заголовке).
Выходные данные: uuid, bot_message.
- **GET /chats**. Входные данные: access_token (в заголовке).
Выходные данные: JSON-файл, содержащий все uuid чатов пользователя.
- **GET /chats/{uuid}**. Входные данные: uuid.
Выходные данные: JSON-файл, содержащий сообщения пользователя и сообщения чат-бота в чате.
- **DELETE /chats/{uuid}**. Входные данные: uuid.
- **wss://chats/{uuid}/generate**. Входные данные: uuid, message. Выходные данные: uuid, response.
- **wss://inference/batching**. Входные данные: uuid, message. Выходные данные: uuid, response, created_at.
- **wss://inference/generate**. Входные данные: batch. Выходные данные: responses, created_at

3. Аппаратные требования.

Сервер для LLM-модели:

- **RAM:** Минимально 16Gb, рекомендуемо 32Gb.
- **Storage:** Минимально 50Gb, рекомендуемо 150Gb.
- **OS:** Linux (Ubuntu/Arch/CentOS).
- **Видеокарта:** Tesla A10, минимально, RTX A5000 24GB рекомендуемо.
- **CPU:** 16–24 ядра с частотой более 2.5 ГГц минимально, 32+ ядер с частотой более 3 ГГц.
- **Скорость сети:** не менее 1 Gbps.

Хранение данных в базе данных

1. **Таблица users.** Таблица содержит информацию о зарегистрированных пользователях. Связь «один-ко-многим» с таблицей chats.

Колонки:

- **id** — уникальный идентификатор пользователя, первичный ключ.
- **username** — уникальное имя пользователя для входа в систему.
- **email** — электронная почта пользователя.
- **password_hash** — хэш пароля, созданный с использованием SCRAM-SHA-256.
- **name** — имя пользователя.
- **surname** — фамилия пользователя.
- **role** — роль пользователя в системе (например: user, admin, moderator).
- **created_at** — дата и время регистрации.

2. **Таблица bot_models.** Таблица содержит справочник доступных LLM-моделей. Связь «один-ко-многим» с таблицей chats.

Колонки:

- **id** — уникальный идентификатор модели, первичный ключ.
- **name** — название LLM-модели.
- **version** — версия модели.
- **is_active** — флаг активности модели.

3. **Таблица chats.**

Таблица содержит информацию о всех созданных чатах. Связь «один ко многим» с таблицами users и bot_models.

Колонки:

- **id** — уникальный идентификатор чата, первичный ключ.
- **uuid** — уникальный идентификатор чата для использования в API.
- **user_id** — идентификатор пользователя, владельца чата.
- **model_id** — идентификатор модели ИИ, используемой в чате.
- **title** — название чата, генерируемое автоматически.

- **created_at** — дата и время создания чата.

4. **Таблица messages.** Таблица содержит информацию о всех сообщениях пользователя и чат-бота в чатах. Связь «многие-к-одному» с таблицей chats.

Колонки:

- **id** — уникальный идентификатор сообщения, первичный ключ.
- **chat_id** — идентификатор чата, которому принадлежит сообщение.
- **role** — роль отправителя: user (пользователь) или assistant (чат-бот).
- **content** — текст сообщения.
- **created_at** — дата и время отправки сообщения.

5. **Таблица message_feedbacks.** Таблица содержит оценки сообщений чат-бота пользователями. Связь «один-к-одному» с таблицей messages, связь «многие-к-одному» с таблицей bot_models.

Колонки:

- **id** — уникальный идентификатор оценки, первичный ключ.
- **model_id** — идентификатор модели, которая сгенерировала ответ.
- **message_id** — идентификатор оцениваемого сообщения от чат-бота.
- **is_positive** — результат оценки: TRUE (лайк), FALSE (дизлайк).
- **created_at** — дата и время оценки.

6. **Таблица user_sessions.** Таблица содержит информацию о активных сессиях пользователей. Связь «многие-к-одному» с таблицей users.

Колонки:

- **id** — уникальный идентификатор сессии, первичный ключ.
- **user_id** — идентификатор пользователя.
- **refresh_token** — токен для обновления access token.
- **expires_at** — срок действия refresh token.
- **created_at** — дата и время создания сессии.

7. Таблица command_suggestions. Таблица содержит часто используемые команды для подсказок пользователям.

Колонки:

- **id** — уникальный идентификатор подсказки, первичный ключ.
- **short_description** — краткое описание команды.
- **command_example** — пример использования команды.
- **usage_count** — счетчик использования для сортировки по популярности.

8. Таблица system_metrics. Таблица содержит метрики производительности и мониторинга системы.

Колонки:

- **id** — уникальный идентификатор метрики, первичный ключ.
- **metric_type** — тип метрики.
- **metric_value** — значение метрики.
- **endpoint** — API endpoint, к которому относится метрика.
- **created_at** — дата и время фиксации метрики.

9. Таблица audit_logs. Таблица содержит журнал аудита действий пользователей и системы. Связь «многие-к-одному» с таблицей users.

Колонки:

- **id** — уникальный идентификатор записи, первичный ключ.
- **user_id** — идентификатор пользователя (может быть NULL).
- **action** — описание действия.
- **ip_address** — IP-адрес, с которого выполнено действие.
- **user_agent** — данные браузера пользователя.
- **timestamp** — дата и время действия.

10. Таблица feedback_analysis. Таблица содержит анализ оценок ответов чат-бота для дообучения моделей.

Колонки:

- **id** — уникальный идентификатор анализа, первичный ключ.
- **message_id** — идентификатор сообщения чат-бота.

- **model_id** — идентификатор модели, сгенерировавшей ответ.
- **analysis_note** — примечание анализа о причине оценки.
- **created_at** — дата и время анализа.