

## **Промежуточный отчет по программному проекту**

### **1. Основные планы и этапы проекта**

1.1 Краткое описание проекта: Проект направлен на озвучивание текстовых сообщений в Telegram-чатах для удобного потребления информации на ходу. Система использует базу данных предыдущих голосовых сообщений отправителей для синтеза голоса, максимально схожего с их реальным голосом и манерой речи. Это позволяет имитировать живое общение и облегчает взаимодействие в мессенджере.

Название проекта: Озвучивание текстовых сообщений в Telegram

Цель проекта: Создать систему, которая преобразует текстовые сообщения в Telegram-чатах в голосовые, имитируя голос и манеру речи отправителей для более естественного восприятия информации.

Краткое описание задач:

А) Найти библиотеки для синтеза голоса на основе данных из базы предыдущих голосовых сообщений.

Б) Интегрировать систему с Telegram для автоматического озвучивания входящих текстовых сообщений.

В) Создать базу данных для хранения голосовых сообщений отправителей и их предпочтений.

Г) Разработать интерфейс бота для настройки и управления функциями озвучивания.

1.2 Планы и этапы выполнения проекта:

Этап проекта	Описание работ	Ожидаемые результаты	Сроки выполнения
Анализ требований	Сбор требований к функционалу, определение целевой аудитории, анализ аналогов.	Техническое задание с описанием всех функций и требований.	1 неделя
Исследование технологий	Выбор технологий для реализации бота: библиотек, TTS-сервисов.	Подробное описание выбранных технологий и инструментов.	1 неделя
Проектирование архитектуры	Разработка архитектуры системы, включая обработку текстов, хранение данных и голосовой синтез.	Схема архитектуры бота, включающая описание всех компонентов.	1 неделя
Создание пользовательских сценариев	Определение взаимодействия пользователей с ботом: обработка команд, уведомления, соглашение.	Сценарии взаимодействия, проработанные для всех случаев использования.	1 неделя
Разработка базового функционала	Реализация базового функционала: обработка команд, преобразование текста в голос.	Рабочий бот с минимальными функциями.	2 недели
Интеграция TTS-сервиса	Подключение Coqui TTS или другого подходящего инструмента синтеза речи.	Полноценная озвучка текстовых сообщений в голос.	1 неделя
Разработка интерфейса	Настройка удобного интерфейса	Удобный и понятный интерфейс с минимальным	1 неделя

	взаимодействия с ботом (inline-режим, кнопки).	количеством команд.	
Разработка системы хранения данных	Реализация базы данных для хранения голосов пользователей, параметров и истории сообщений.	Рабочая база данных с безопасным доступом и интеграцией в бота.	2 недели
Тестирование базового функционала	Проверка работы бота в личных и групповых чатах, обработка ошибок и багов.	Исправленный и стабильный бот.	1 неделя
Разработка расширенного функционала	Добавление кастомизации голоса (тембр, скорость), поддержка отправки файлов, генерация диалогов.	Функционал с модификацией голосов и поддержкой дополнительных функций.	2 недели
Тестирование системы	Полное тестирование всех функций, включая нагрузочные тесты и обработку исключений.	Устойчивый к нагрузкам и безошибочный бот.	2 недели
Подготовка к запуску	Подготовка пользовательского соглашения, настройка логирования, развертывание на сервере.	Готовый к использованию бот с документированным функционалом.	1 неделя
Развертывание и запуск	Установка на сервер, запуск бота для пользователей.	Бот доступен для пользователей.	1 неделя
Сопровождение и обновление	Сбор отзывов пользователей, внесение	Обновления и улучшения	Постоянно

	изменений, добавление новых функций.	функционала на основе отзывов.	
--	--	-----------------------------------	--

## 2. Используемый технологический стек и его обоснование

### 2.1 Перечень используемых технологий:

Технология/Инструмент	Описание	Причины выбора
Python	Язык программирования, используемый для написания серверной части Telegram-бота.	Простота и удобство для разработки, широкая экосистема библиотек, поддержка асинхронности.
aiogram	Библиотека Python для взаимодействия с Telegram API.	Асинхронная архитектура, удобство работы с ботами, поддержка современных возможностей Telegram API.
Coqui TTS	Библиотека для синтеза речи на основе нейронных сетей.	Высокое качество синтеза речи, открытый код, поддержка кастомных моделей.
PostgreSQL	Реляционная база данных для хранения данных пользователей, голосов и истории сообщений.	Надёжность, высокая производительность, поддержка сложных запросов.
psycopg2	Библиотека Python для асинхронного взаимодействия с PostgreSQL.	Высокая производительность, отличная интеграция с асинхронными приложениями на Python.
Docker	Платформа для контейнеризации приложений.	Упрощение развёртывания и масштабирования, обеспечение изоляции окружения.

## 2.2 Обоснование выбранного технологического стека:

Выбранный стек технологий был тщательно подобран для достижения оптимального баланса между производительностью, удобством разработки, масштабируемостью и безопасностью.

Библиотека `aiogram` позволяет быстро и эффективно создавать функционал бота благодаря своей асинхронной архитектуре, полной поддержке Telegram API и удобству работы с командами и inline-режимами.

Soqui TTS была выбрана как инструмент для синтеза речи, поскольку это одна из самых мощных и гибких библиотек с поддержкой мультязычности, включая русский язык, что является ключевым требованием для проекта. Библиотека открыта для сообщества, не имеет жёстких лицензионных ограничений, а также поддерживает кастомизацию голосов, что позволяет создать максимально естественные голосовые сообщения. Возможность подключения через API упрощает её интеграцию в архитектуру бота.

Для хранения данных выбрана PostgreSQL благодаря её надёжности, высокой производительности и поддержке сложных запросов. Это позволяет эффективно хранить и обрабатывать базу голосов пользователей, параметры их взаимодействий и историю сообщений. Для асинхронного взаимодействия с базой используется `psycopg2`, что обеспечивает минимальные задержки при работе с запросами.

Docker был включён в стек для контейнеризации приложения, что обеспечивает лёгкость развертывания и переносимость между различными средами. Это также позволяет изолировать окружение и минимизировать возможные конфликты между зависимостями.

## 3. Критерии оценивания проекта

В зависимости от курса и уровня студента, сформируйте, пожалуйста, перечень количественных критериев (не менее 3 критериев),

которыми будет оцениваться проект. Вы можете предложить и обосновать свои критерии, либо использовать такие критерии как, например:

Критерий	Описание
Использование инструмента автоматической документации API	Будут использованы
Работа с базой данных	Будут использованы
Функциональность - Процент выполнения функциональных требований	Выполненные требования в процентах от общего количества
Производительность и эффективность - Время отклика	Среднее время ответа программы на запросы пользователя (в секундах)
Качество кода - Покрытие кода тестами (%)	Процент строк кода, покрытых тестами
Документация и оформление - Полнота документации (%)	Процент от требуемого объема документации
Тестирование - Количество написанных тестов	Общее количество юнит-тестов, интеграционных тестов и других видов тестов
Тестирование - Процент успешных тестов (%)	Процент успешно пройденных тестов из общего количества
Соблюдение сроков и плана - Процент выполнения работы в срок (%)	Процент задач, выполненных в срок
Использование технологического стека - Процент использования функциональности стека (%)	Процент использования функциональности выбранного стека технологий

#### 4. Особые пометки

Здесь указываются любые особые примечания, риски, изменения в планах, а также другие важные детали, которые необходимо учесть на данном этапе проекта.