

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**  
**«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ**  
**УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»**

Институт компьютерных наук и технологического образования

Кафедра компьютерных технологий и электронного обучения

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Разработка диалоговой системы для автоматического расчёта  
калорийности и макронутриентного состава рациона

Направление подготовки: «Информатика и вычислительная техника»

Руководитель:

доцент, кандидат технических  
наук,

\_\_\_\_\_ Карпова Н.А.

«\_\_»\_\_\_\_\_ 2024 г.

Автор работы:

Студент 4 курса 1 группы

\_\_\_\_\_ Помыкин М.А.

«\_\_»\_\_\_\_\_ 2024 г.

## **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

на разработку программного продукта

"Диалоговая система для автоматического расчёта калорийности и макронутриентного состава рациона"

[Обозначение: ТЗ-ВКР-2026]

Разработчик: Помыкин Михаил Антонович

Заказчик: кафедра института

Дата: 08.12.2025

### **1. НАИМЕНОВАНИЕ И ОБОЗНАЧЕНИЕ РАБОТЫ**

Разработка диалоговой системы автоматического расчёта калорийности и макро нутриентного состава рациона питания в формате Telegram-бота с веб-приложением (Mini App).

### **2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ**

Выполнение требований к выпускной квалификационной работе в рамках Федерального государственного образовательного стандарта по направлению подготовки [09.03.02 Информатика и вычислительная техника].

### **3. НАЗНАЧЕНИЕ РАЗРАБОТКИ**

Автоматизация процесса учёта калорийности и макронутриентов (белки, жиры, углеводы) рациона питания пользователей через Telegram-интерфейс с парсингом естественного языка и веб-приложением для аналитики потребления.

## 4. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

### 4.1 Функциональные требования

- **4.1.1 Парсинг естественного языка:** Автоматическое извлечение из сообщения вида «Куриная грудка 200г» → название продукта, вес порции.
- **4.1.2 База продуктов:** >5000 позиций с КБЖУ на 100г (USDA + российские таблицы).
- **4.1.3 Расчёт порций:** Автоматический пересчёт КБЖУ на указанный вес.
- **4.1.4 Хранение данных:** История приёмов пищи по Telegram ID пользователя.
- **4.1.5 Telegram Mini App:** Календарь, списки приёмов пищи, суточные/недельные отчёты, редактирование записей.
- **4.1.6 Экспорт данных:** PDF/Excel отчёты.

### 4.2 Требования к надёжности

- Graceful shutdown сервисов.
- Retry логика для Telegram Bot API (3 попытки, backoff 1-2-4с).
- Логирование в Loki (structured logging slog).
- Резервное копирование БД (pg\_dump cron).

### 4.3 Требования к интерфейсу

- Telegram Bot API v7.0+.
- Telegram Mini App (WebView) совместим с Desktop/Mobile iOS/Android.
- Responsive дизайн для экранов 320-1920px.

4.4 Требования к производительности

- Время ответа бота < 2 сек (p95).
- Поддержка 1000 одновременных пользователей.
- БД: 1000 RPS чтение, 100 RPS запись.

4.5 Требования к безопасности

- Изоляция данных по Telegram ID.
- Защита от SQL-инъекций (prepared statements).
- Rate limiting (10 запросов/мин на пользователя).
- HTTPS для Mini App.

4.6 Требования к совместимости

- Telegram: iOS 14+, Android 7+, Desktop.
- Браузеры Mini App: Chrome 90+, Safari 14+, Firefox 88+.

4.7 Аппаратные требования

Компонент	CPU	RAM	Storage
Backend (Go)	2 vCPU	2 GB	-
PostgreSQL	2 vCPU	4 GB	50 GB SSD
Frontend (Vue)	1 vCPU	1 GB	-

Таблица 1 “ Аппаратные требования”

5. ТРЕБОВАНИЯ К СОСТАВУ И СВОЙСТВАМ ПРОДУКЦИИ

- **Backend:** Go 1.22+, PostgreSQL 17, Docker, Kubernetes, Loki.

- **Frontend:** Vue 3 / React 18 (Telegram Mini App).
- **Инфраструктура:** Docker Compose (dev), Kubernetes (prod).
- **Мониторинг:** Loki + Grafana.

## 6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Показатель	До автоматизации	После автоматизации
Время ввода приёма пищи	2-3 мин	5 сек
Точность расчётов КБЖУ	70-80% (ручной ввод)	95%+ (база данных)
Кол-во продуктов в БД	-	5000+ позиций

Таблица 2 “ Техничко-экономическис покарзатели”

Экономический эффект: сокращение трудозатрат на 95% при ведении дневника питания.

## 7. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

- Руководство пользователя (Markdown/PDF).
- Руководство администратора (деплой K8s, мониторинг Loki).
- API документация (OpenAPI 3.0 / Swagger).
- Описание архитектуры (диаграммы C4 model).
- Описание алгоритмов парсинга естественного языка.

Ниже вариант раздела «Сокращения» для вставки в  
T3.ISR-1.1-Tekhnicheskoe-zadanie.docx

## Сокращения

- КБЖУ — калорийность, белки, жиры, углеводы
- БД — база данных.
- API — программный интерфейс приложения (Application Programming Interface)
- PDF — формат переносимого документа (Portable Document Format)
- SQL — язык структурированных запросов (Structured Query Language)
- RPS — число запросов в секунду (Requests Per Second)
- E2E — сквозное тестирование (End-to-End)
- K8s — система оркестрации контейнеров Kubernetes
- CI/CD — непрерывная интеграция и доставка (Continuous Integration / Continuous Delivery)ъ
- HTTPS — защищённый протокол передачи гипертекста (HyperText Transfer Protocol Secure)

## 8. ЭТАПЫ И СТАДИИ РАЗРАБОТКИ

№	Этап	Содержание работ	Срок
1	Анализ и проектирование	БД схема, API спецификация, архитектура	01.09.2025-30.09.2025
2	Разработка backend	Go сервисы, PostgreSQL, парсинг, API	01.10.2025-31.01.2026
3	Разработка frontend	Telegram Bot, Vue Mini App	01.02.2026-28.02.2026
4	Инфраструктура	Docker, K8s, Loki, CI/CD	01.03.2026-15.03.2026
5	Тестирование	Unit/Integration/E2E, нагрузка	16.03.2026-31.03.2026
6	Документация и приёмка	Финальная документация, предзащита	01.04.2026-15.04.2026

## 9. ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ

### 9.1 Средства контроля:

- Unit-тесты: покрытие >80% (go test).
- Интеграционные тесты: API endpoints.
- E2E тесты: Telegram Bot сценарии.
- Нагрузочное тестирование: k6 (1000 RPS).

## **9.2 Критерии приёмки:**

- Все функциональные требования реализованы.
- Тесты пройдены, покрытие >80%.
- Документация сдана в полном объёме.
- Демонстрация работы: живой бот + Mini App.

## **9.3 Порядок приёмки:**

1. Предварительная приёмка научным руководителем.
2. Финальная защита перед государственной экзаменационной комиссией.