

АИС "Школьное питание" - Техническая документация

**Автоматизированная информационная
система "Школьное питание"**

**Командный кейс №2 "Управление
столовой"**

Команда: Оранжевая

Участники:

Школа: Л2Ш

Руководитель: Кирилл Евгеньевич

Краткое описание проекта

Веб-приложение на Flask для полной автоматизации работы школьной столовой. Система заменяет бумажные журналы и талоны цифровым решением, предоставляя ученикам — интерактивное меню и заказ, поварам — систему учета блюд и инвентаря, администрации — финансовую аналитику и отчетность. Все процессы от заказа до получения пищи объединены в единый цифровой контур.

Ключевые особенности нашей реализации

3.1. Встроенная платежная система с виртуальным балансом

Каждый пользователь имеет внутренний счет (поле "money" в профиле)

Пополнение через интерфейс системы без внешних платежных шлюзов

Автоматическая проверка достаточности средств при оформлении заказа

Полная финансовая прозрачность — все транзакции фиксируются

3.2. Полный цикл данных в реальном времени

Заказ ученика → автоматическое попадание к повару → отметка о готовности → подтверждение получения

Каждый статус фиксируется в системе: "ожидание", "готово", "получено"

Данные синхронизируются между всеми участниками без задержек

3.3. Система безопасности с подтверждением регистрации
Обязательное подтверждение email через одноразовый код
(SMTP или консоль в режиме отладки)
Сессии Flask с защищенными cookies (HTTPS only,
HTTPOnly, SameSite=Lax)
Строгая проверка прав доступа для каждого маршрута
Валидация всех входящих данных

3.4. Управление инвентарем с каскадным согласованием
Реальный учет остатков продуктов с единицами измерения
(кг, л, шт)
Визуализация остатков продуктов через прогресс-бары в
интерфейсе повара

3.5. Готовая отчетность с генерацией Excel-файлов
Встроенный генератор отчетов на базе библиотеки Pandas
Два типа отчетов: по заказам учеников и по закупкам
продуктов
Автоматическое форматирование, настройка ширины
колонок
Именование файлов с timestamp для архивирования

3.6. Умная система фильтрации и аллергенов
Индивидуальный профиль аллергий для каждого ученика
Автоматическое скрытие опасных блюд при включенном
фильтре
Визуальное выделение аллергенов в карточках товаров
Динамическая фильтрация без перезагрузки страницы
(JavaScript)
Технологический стек и обоснование выбора

4.1. Backend: Python + Flask
Flask выбран как легковесный и гибкий фреймворк

Быстрая разработка прототипа с минимальными накладными расходами
Простота развертывания и отладки

4.2. Frontend: HTML5 + CSS3 + JavaScript + Bootstrap 5
Bootstrap для адаптивного дизайна и готовых компонентов
Чистый JavaScript для динамической фильтрации
Jinja2 для шаблонизации и передачи данных

4.3. Хранение данных: JSON-файлы
Отсутствие зависимости от внешних СУБД
Простота резервного копирования и восстановления
Прозрачность структуры данных для разработчиков
Легкость отладки и тестирования

4.4. Библиотеки и инструменты:
Flask-Session для управления сессиями
Pandas + Openpyxl для генерации Excel-отчетов
SMTP-библиотека для отправки email
Git для контроля версий
Архитектура и схема работы данных

5.1. Ролевая модель:
Ученик (права=1): заказ, оплата, получение
Повар (права=2): приготовление, учет инвентаря, заявки на закупку
Администратор (права=3): согласование заявок, отчетность, статистика

5.2. Поток данных при заказе:
Ученик добавляет блюда в корзину (кнопки +/-)
Система проверяет баланс при оформлении
При успехе: создается запись в student_to_povar.json

Повар видит заказ в своей панели, отмечает "готово"

Заказ появляется у ученика с кнопкой "получить"

После получения: статус фиксируется в student_buys.json

5.3. Поток данных при закупке:

Повар видит недостаток продукта на складе

Создает заявку через форму → pовар_to_admin.json

Администратор видит заявку, согласовывает или отклоняет

Статус обновляется, повар видит результат

7.1. Требования:

Python 3.8 или выше

Установленный pip

7.2. Шаги:

Клонировать репозиторий:

git clone [ссылка_на_репозиторий]

cd school_dining

Установить зависимости:

pip install flask flask-session pandas openpyxl

Создать структуру папок (выполняется автоматически при первом запуске)

Запустить приложение:

python app.py

Открыть в браузере:

<http://127.0.0.1:5237>

7.3. Тестовые аккаунты:

Ученик: email: student@student, пароль: student

Повар: email: povar@povar, пароль: povar

Администратор: email: admin@admin, пароль: admin

8.2. Видео-демонстрация работы системы:

[Вставьте ссылку на YouTube/RuTube/другой хостинг]

Соответствует регламенту испытаний, демонстрирует весь функционал.

Заключение

Система представляет собой законченное решение для автоматизации школьной столовой. Все ключевые процессы оцифрованы: от заказа до финансовой отчетности. Особенность реализации — самостоятельность системы (внутренний баланс, хранение в JSON, генерация отчетов) без зависимостей от внешних сервисов.

Архитектура позволяет легко расширять функционал и адаптировать систему под нужды конкретной школы.

Система готова к использованию и демонстрирует полный цикл автоматизации питания в образовательном учреждении.