**1. Краткое описание проекта (суть)**

Название мобильного приложения: Система управления заказами пиццерии (Pizza Order Management System)

Описание: Веб-приложение для автоматизации процессов приема, обработки и отслеживания заказов в пиццерии. Включает ролевую модель (администратор, клиент), управление меню, корзину, оформление заказов и отслеживание их статусов в реальном времени.

Цель: Повышение эффективности работы пиццерии за счет автоматизации процессов приема и обработки заказов, уменьшения ошибок ручного ввода, улучшения клиентского опыта.

Аналоги: iiko, Presto, R-Keeper.

Дата сдачи — **8 декабря 2025г.**

**2. Основные требования (Функциональные)**

**2.1. Пользовательские роли**

* **Клиент:** регистрация/вход, оформление заказа, просмотр меню, корзина, отслеживание статуса заказа.
* Администратор: управление меню, просмотр и изменение статусов заказов, экспорт данных.

**2.2. Базовый функционал (MVP)**

1. Аутентификация: раздельный вход для клиента (по контактным данным) и администратора (логин/пароль).
2. Управление меню: добавление, редактирование, удаление пицц (название, цена, изображение).
3. Корзина: добавление/удаление пицц, расчет стоимости.
4. Оформление заказа: ввод комментария, сохранение заказа в БД.
5. Отслеживание заказов: клиент видит статус своего заказа; администратор видит все заказы и меняет их статусы («Готовится», «В пути», «Приехало», «Завершен»).
6. Экспорт данных: выгрузка меню в Excel.
7. Валидация данных: проверка корректности ввода (телефон, обязательные поля).

**2.3. Нефункциональные требования**

* Платформа: Desktop (Windows/Linux/macOS) на базе Python + Tkinter.
* БД: SQLite.
* Производительность: отклик интерфейса < 1 сек на основные операции.
* Безопасность: валидация ввода, защита от SQL-инъекций.
* Удобство использования: интуитивный интерфейс, понятные статусы заказов.

**3. Предложенный стек технологий (рекомендация для сдачи в сроки)**

- Frontend: Tkinter (Python)

- Backend: Python (логика приложения)

- БД: SQLite

- Экспорт: pandas + openpyxl для Excel

- Обоснование: простота разработки, минимальные зависимости, подходит для учебного проекта.

**4. Рекомендуемые инструменты разработки**

* IDE: VSCode / Android Studio.
* Figma — прототипы / дизайн.
* GitHub — исходники + Actions.
* Firebase Console — backend.
* Flutter DevTools — отладка.
* Postman / REST Client — если есть API.

**5. Архитектура и данные**

**5.1. Архитектурная схема (вкратце)**

Однокомпонентное приложение (монолит) с слоем UI (Tkinter), бизнес-логикой и слоем данных (SQLite).

**5.2. Структура БД**

* pizzas (id, name, price, image\_path)
* users (id, name, phone, address) — клиенты
* admins (id, login, password) — администраторы
* orders (id, user\_id, comment, status, total\_price, created\_at)
* order\_items (id, order\_id, pizza\_id, quantity)

**6. API / Endpoints (пример для собственного backend)**

Не применяется

**7. UI/UX: основные экраны + требования**

1. Окно авторизации (выбор роли: клиент/админ)

2. Меню для клиента – список пицц с кнопкой «В корзину»

3. Корзина – список выбранного, итоговая сумма, кнопка «Заказать»

4. Окно заказа – ввод комментария, подтверждение

5. Окно статуса заказа – текущий статус, история (для клиента)

6. Панель администратора – управление меню, просмотр заказов, смена статусовWireframe: минималистичный, карточный интерфейс; крупные кнопки, читаемые списки.

**8. Тестирование**

* Модульные тесты: логика корзины, расчет суммы, валидация.
* Интеграционные тесты: оформление заказа, изменение статуса.
* Ручное тестирование: UI, сценарии использования.

**9. Релиз**

* Репозиторий: GitHub
* Release: Google Play / RuStore.

**10. Безопасность и приватность**

* Валидация ввода (телефон, email если есть)
* Хеширование паролей админов (например, через bcrypt)
* Защита от SQL-инъекций (параметризованные запросы)
* Возможность удаления аккаунта / данных.

**11. Роадмап / план на 4 недели**

Неделя 1:

* Проектирование БД
* Реализация окон авторизации и меню
* Настройка SQLite

Неделя 2:

* Корзина и оформление заказа
* Панель администратора (управление меню)

Неделя 3:

* Отслеживание заказов
* Смена статусов
* Экспорт в Excel

Неделя 4:

* Тестирование, отладка
* Документация
* Подготовка к сдаче

**12. Уникальные функции проекта (если останется время)**

* Уведомления клиенту на email/SMS
* История заказов для клиента
* Система скидок/промокодов
* Интеграция с картами (для доставки)
* Пуш-уведомления с настройками (типы уведомлений).
* Темная тема.