Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Смоленский государственный университет

Физико-математический факультет

Кафедра прикладной математики и информатики

**Проект: Приложение SERVERoff для точки питания с узбецкой выпечкой**

Разработкой занимались:

студенты 4 курса

специальности Прикладная математика и информатика

Антонов Дмитрий Юрьевич и Меметов Сервер Асан-Угли

Смоленск, 2025

**Введение**

Цель работы: на основе создания приложения для Android изучить процесс и методы создания мобильного приложения.

Тема работы: основные этапы разработки приложения для Android; нюансы при разработке мобильного приложения.

Для разработки проекта была взята идея сделать приложение для точки питания с узбецкой выпечкой и быстрой едой. Выбранным языком для разработки был выбран Python 3.13, хотя изначально была попытка разработать на C#. За фреймворк GUI был взят Kivy, а работа проводилась в VS Code. Python был взят из-за его гибкости и лёгкого синтаксиса. И Kivy оказался хорошим и красивым решением для данного проекта. Изначально, когда пробовали в C#, то работа с .NET Multi-platform App UI (.NET MAUI) была тоже достаточно интересной. MAIU тоже может составить хорошую графическую составляющую. Но язык C# немного сложнее по синтаксису и часто происходила путаница, например, среди классов и функций. Но это также решалось в основном расширенной структурой проекта. Также использовался ASP.NET. Предполагалось создать сервис и клиент отдельно. Но по мере добавления функций, решения проектов становились громоздкими и тяжело читаемыми для новичков как мы. Работа с Python была лучшим решением для нас.

Вот основные функции, которые изначально планировалось реализовать: Зарегистрироваться как Клиент, Зарегистрироваться как Организация, Войти в Систему как Клиент, Изменить профиль, Изменить профиль другого пользователя, Заблокировать пользователя, Добавить организацию, Дать права на управление организацией, Управлять справочником, Добавить Новость, Изменить график работы, Загрузка/Выгрузка расписания.

А также планировалось дополнительно реализовать следующий функционал: Управление Клиентами, Оплатить за сервис, Получить отчеты по организации, Записаться, Дополнительные функциональные требования, Отправка sms, Форум, Система отзывов и предложений, Изменение контента сайта, Изменение справочников Системы, Логирование операций в Системе, Интеграция с социальными сетями, Интеграция с другими Системами, Отчеты.

В процессе разработки было совершено много попыток, возникали различные ошибки, которые устранялись сразу же. В первую очередь была цель реализовать функционал. Интерфейс был далеко не главным пунктом в этом проекте. Далее приведём структуру, использованные методы, библиотеки и т.д.; работу реализованного функционала.

**Основные методы разработки.**

Как и говорилось ранее, планировалось реализовать некоторый функционал, но в связи с знакомством с разработкой приложения были реализованы лишь небольшое количество функций.

Для клиентов:

* Регистрация/авторизация
* Просмотр меню и продуктов
* Добавление в корзину
* Оформление заказов
* Просмотр истории заказов
* Написание отзывов
* Управление профилем

Для организаций:

* Управление продуктами
* Просмотр заказов
* Статистика продаж
* Управление графиком работы
* Публикация новостей
* Просмотр отчетов

Для администраторов:

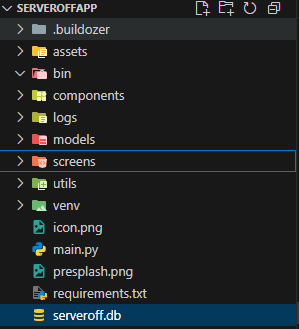
* Управление пользователями
* Блокировка аккаунтов
* Управление организациями
* Настройка системных справочников
* Модерация контента

Были добавлены следующие основные страницы:

* Страница входа
* Страница регистрации пользователя (Клиент)
* Страница регистрации пользователя (Организация)
* Мой профиль
* Настройки
* Новости
* Заказы клиента
* Заказы организации
* Корзина
* График работы
* Социальные сети
* История заказов
* Статистика
* Меню продуктов

С помощью фреймворка Kivy был реализован простой интерфейс с закруглёнными элементами. На главной странице планировалось добавить NavBar. Почти на каждой странице можно заметить ScrollBar. Его присутствие было необходимо, так все элементы не помещаются на одну страницу. Также есть моменты, когда некоторые элементы немного вылезают. Планировалось оптимизировать каждую страницу по любой размер окна.

Теперь о структуре и библиотеках. Сама структура проекта выглядит так:



Основные фреймворки, которые использованы в работе:

* Kivy 2.1.0 - Основной фреймворк для кроссплатформенного UI
* KivyMD 1.1.1 - Material Design компоненты для Kivy
* SQLite3 - Встроенная база данных

Основные таблицы базы данных:

* users - Пользователи системы
* organizations - Организации/рестораны
* products - Товары и меню
* orders - Заказы
* order\_items - Элементы заказов
* reviews - Отзывы и рейтинги
* user\_permissions - Права доступа
* payments - Платежи
* appointments - Записи и бронирования

Ключевые библиотеки Python:

* Pillow==9.5.0 - Обработка изображений
* requests==2.28.2 - HTTP запросы
* hashlib - Хеширование паролей
* logging - Система логирования
* datetime - Работа с датами и временем
* sqlite3 - База данных

Основные UI компоненты:

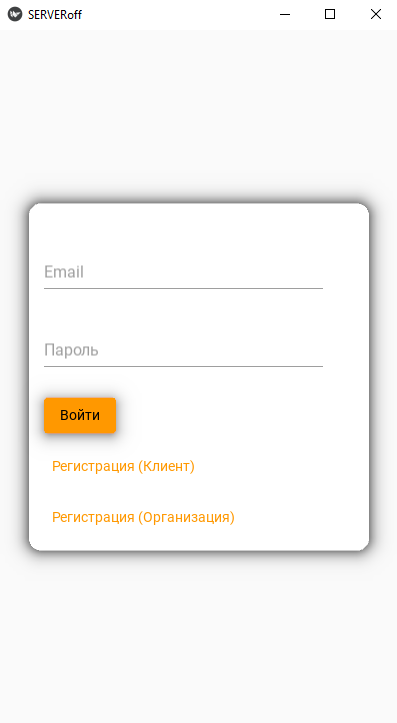
* MDScreenManager - Управление экранами
* MDTopAppBar - Верхняя панель
* MDCard - Карточки
* MDRaisedButton - Кнопки
* MDTextField - Поля ввода
* MDLabel - Текстовые метки
* MDDialog - Диалоговые окна
* MDScrollView - Прокручиваемые области

Основные экраны приложения:

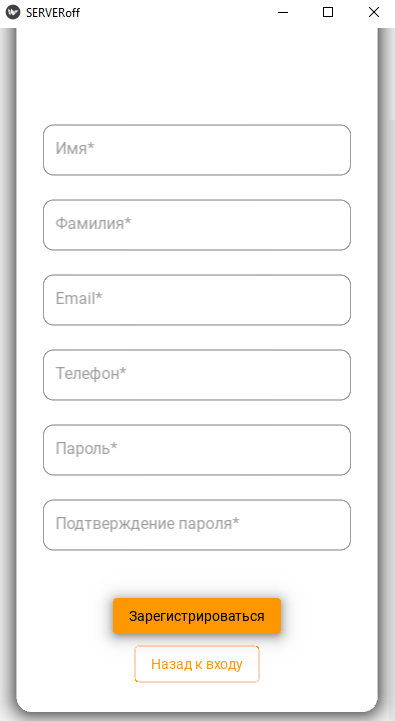
* LoginScreen() - Вход в систему
* RegisterScreen() - Регистрация
* DashboardScreen() - Главная панель
* ProductsScreen() - Управление продуктами
* OrdersScreen() - Просмотр заказов
* CartScreen() - Корзина покупок
* ProfileScreen() - Профиль пользователя
* SettingsScreen() - Настройки
* AdminUsersScreen() - Управление пользователями (админ)

**Скриншоты работы приложения.**

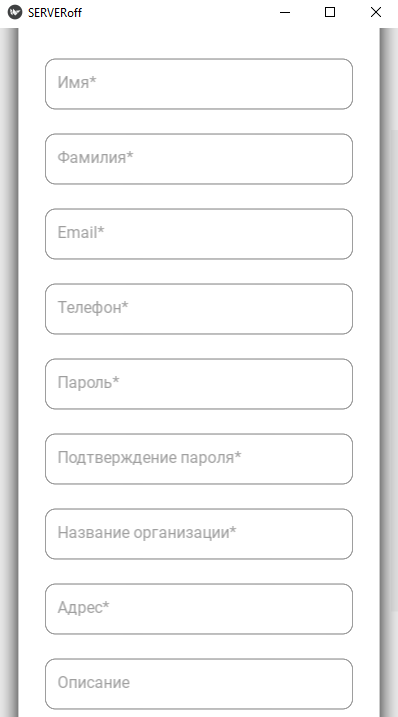
Страница входа.



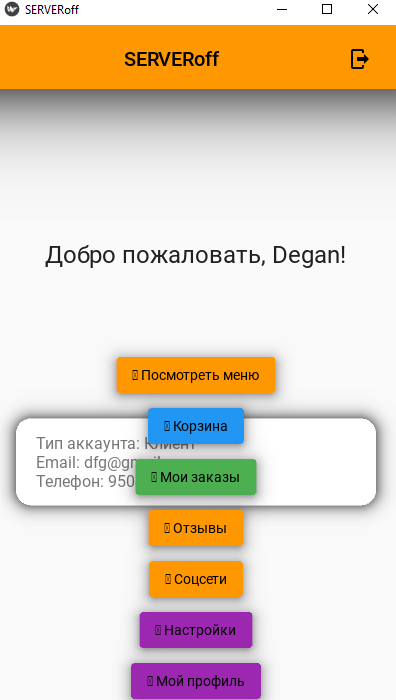
Страница регистрации клиента.



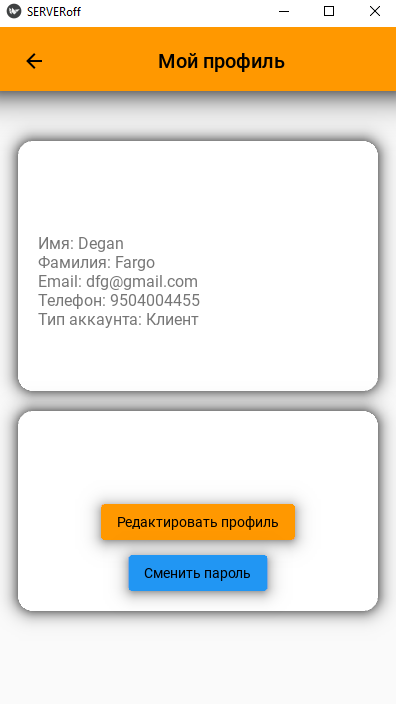
Страница регистрации организации.



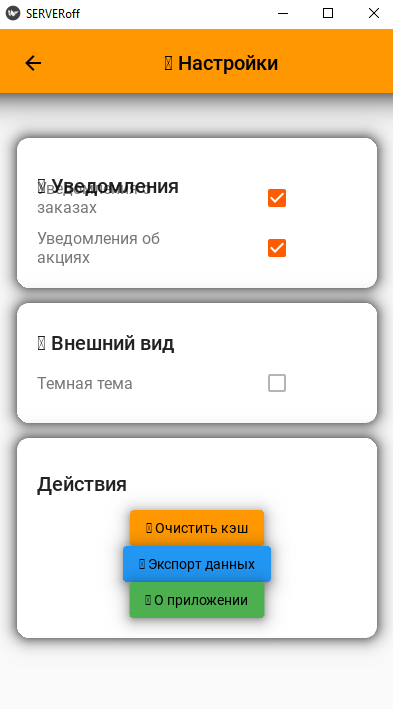
Главная страница клиента.



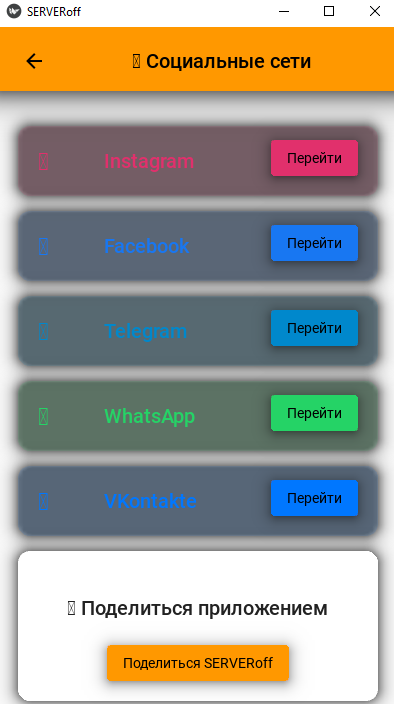
Мой профиль.



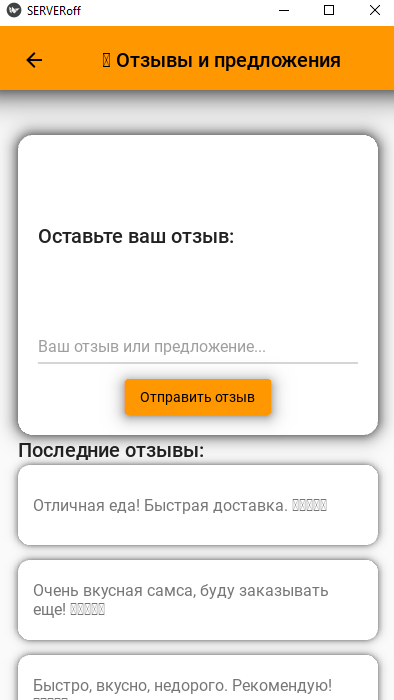
Настройки.



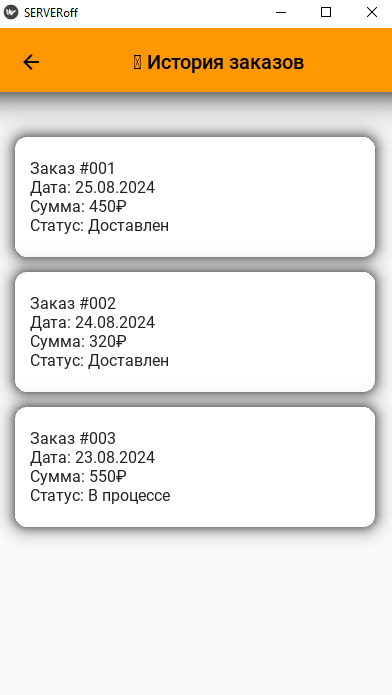
Социальные сети.



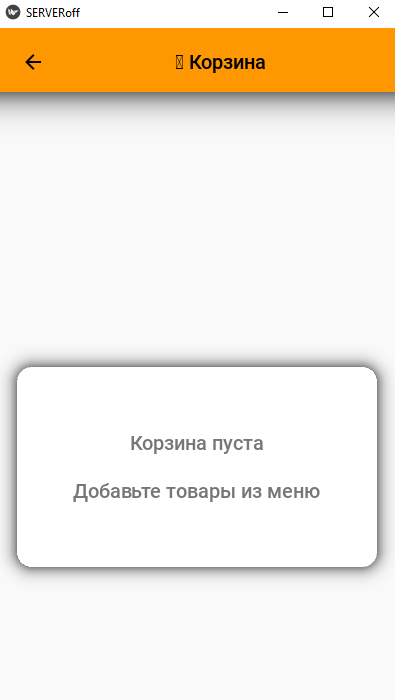
Отзывы и предложения.



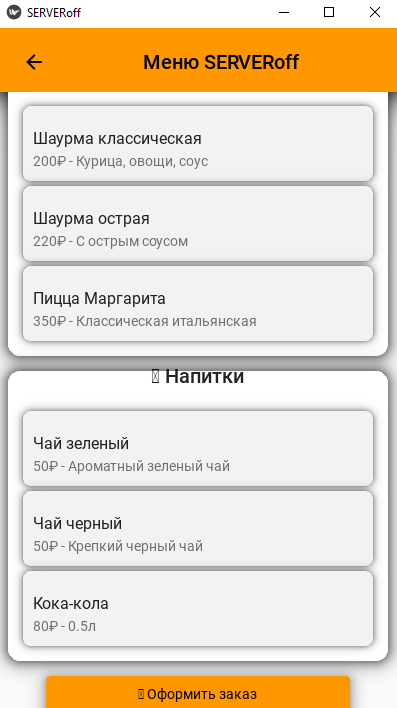
История заказов.



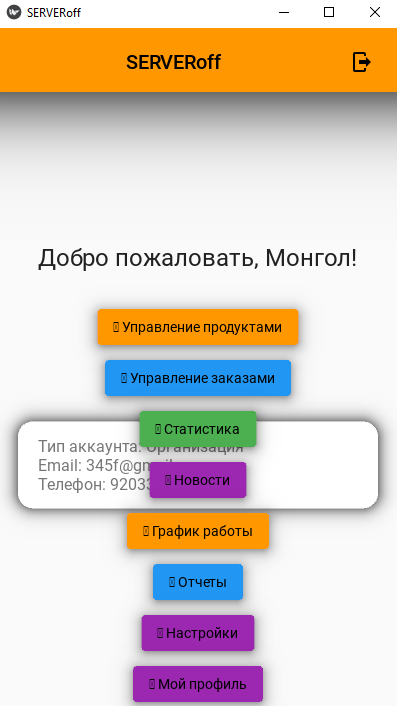
Корзина.



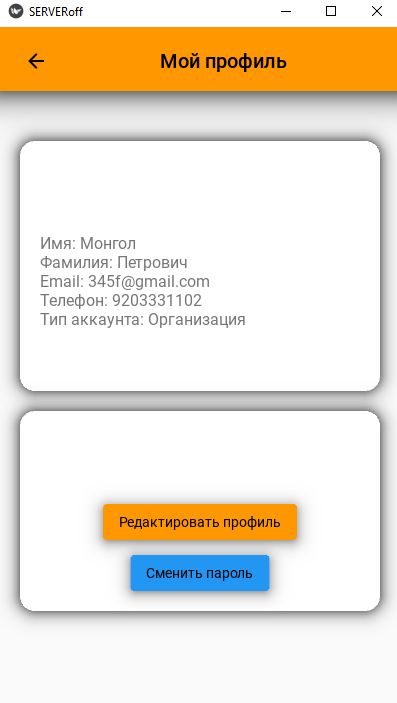
Меню.



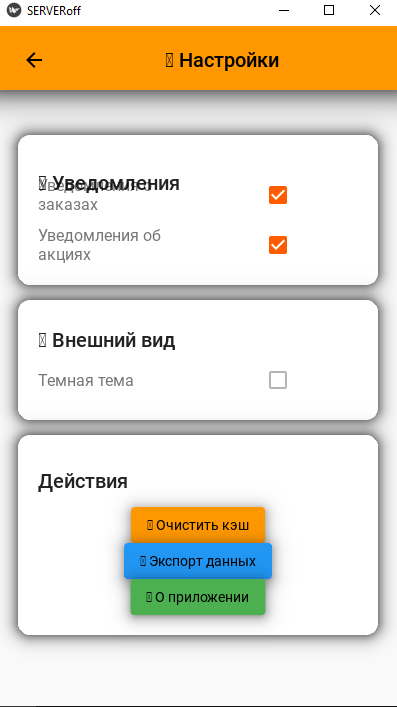
Главная страница организации.



Мой профиль.



Настройки.



Отчёты.

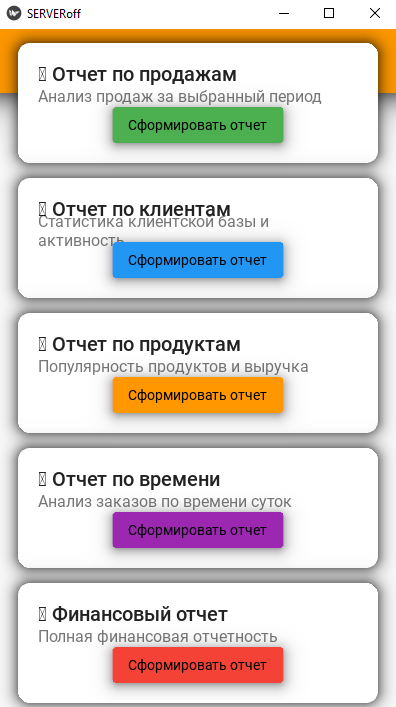
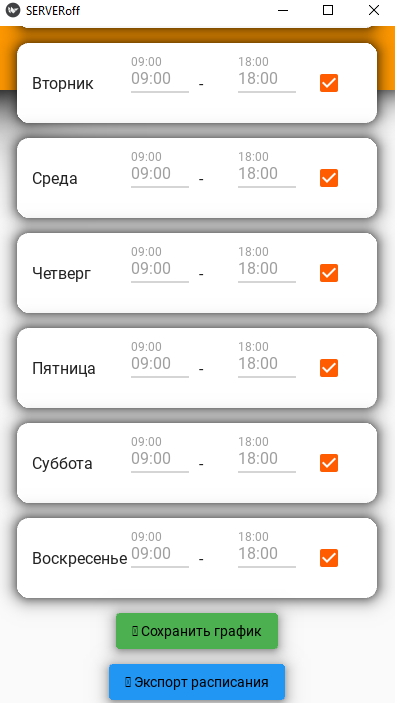
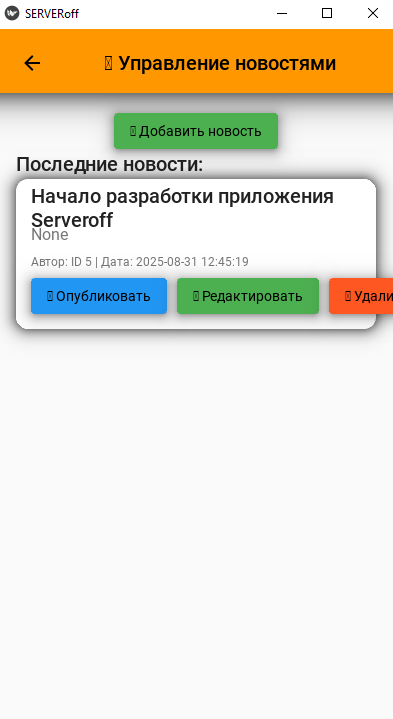


График работы.



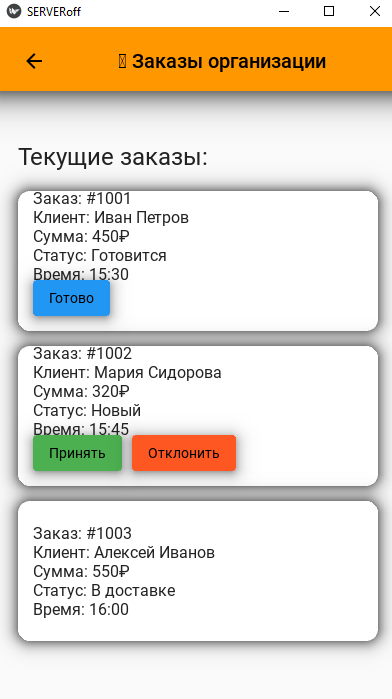
Новости.



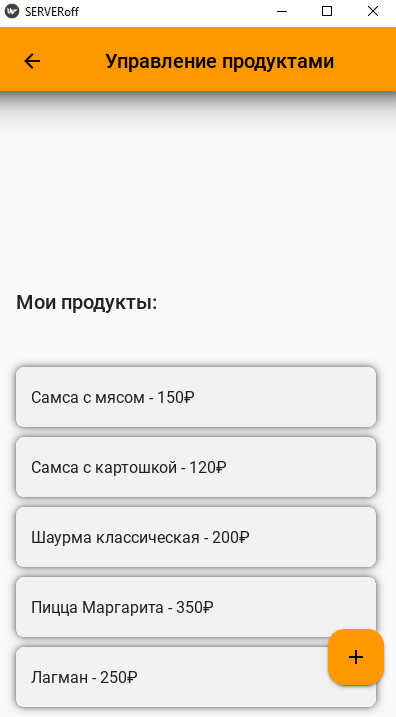
Статистика.



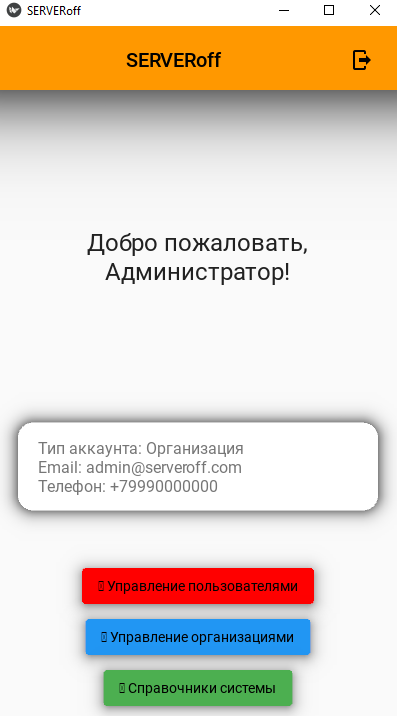
Заказы организации.



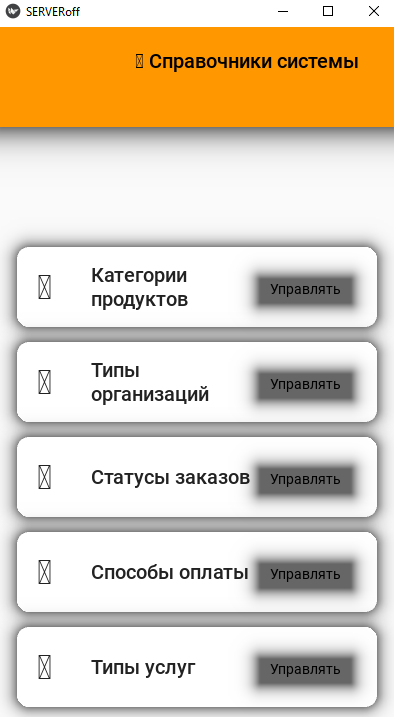
Продукты.



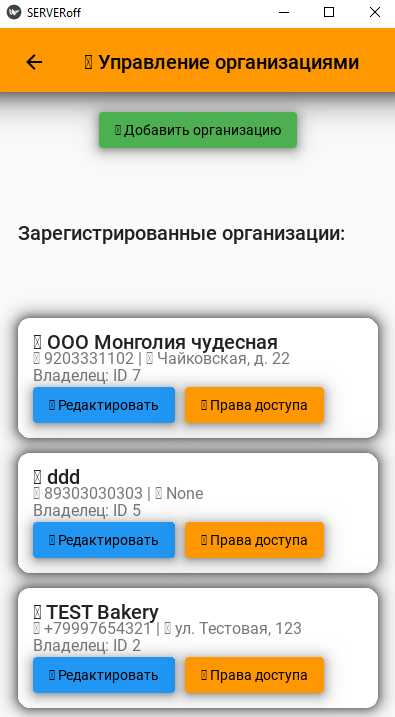
Главная страница администратора.



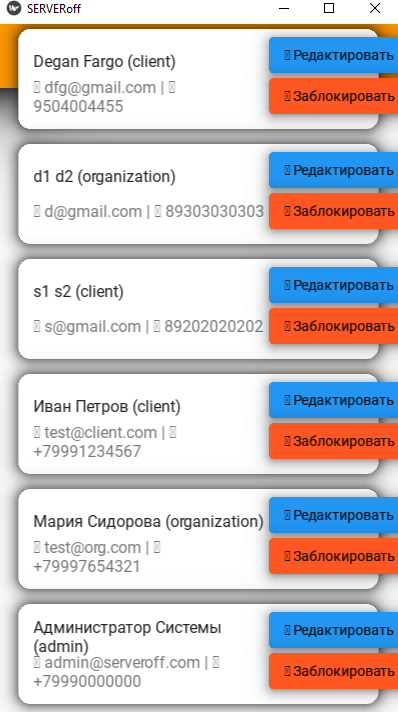
Справочники системы.



Управление организациями.



Управление пользователями.



**Заключение**

Разработка SERVERoff стала комплексным введением в мир кроссплатформенной разработки на Python. Мы начали с чистого листа и прошли весь путь от концепции до макета приложения, оснащенного базовыми функциями для пищевого бизнеса. Главным моментом стала связка Kivy и KivyMD, которые позволили создать современный Material Design интерфейс с закругленными углами и светлой темой.

На практике мы освоили архитектурный подход с разделением на модули: ядро системы в utils, экраны приложения в отдельной папке, что обеспечило поддерживаемость кода. Реализация многоуровневой системы прав доступа показала важность продуманной безопасности — мы внедрили хеширование паролей, проверки прав доступа и логирование операций. Работа с SQLite научила нас проектировать реляционную структуру базы данных с связями между пользователями, организациями и заказами.

Самой значительной трудностью оказалась сборка под Android — мы столкнулись с необходимостью настройки Buildozer и совместимости библиотек, что потребовало глубокого погружения в инструменты сборки. Создание адаптивного интерфейса, который выглядел бы одинаково хорошо на разных устройствах, также потребовало множества итераций и тонкой настройки layout-менеджеров.

Этот проект показывает, что Python может быть эффективным инструментом для создания полноценных мобильных приложений со сложной бизнес-логикой. Мы создали приложение приложение, которое имеет различные пути развития. Опыт, полученный при разработке SERVERoff, будет очень полезен для будущих проектов в области мобильной разработки.