

## ПРИЛОЖЕНИЕ ДЛЯ СОСТАВЛЕНИЯ СПИСКА ПОКУПОК НА ОСНОВЕ ВЫБРАННЫХ РЕЦЕПТОВ

Составление списка покупок – это всегда долгий и не самый интересный процесс. Особенно, когда дело касается закупок продуктов к большим праздникам, например, Новый Год. Необходимо запланировать покупку разных продуктов в нужном количестве, сверяясь с книгой рецептов. Возникла идея разработать мобильное приложение для людей, которые часто готовят различные блюда по рецептам и, как следствие, часто ходят в магазин.

Приложение объединяет в себе книгу рецептов и списки покупок. Реализованы две роли: обычного пользователя и модератора. Для пользователя приложения реализованы следующие функции:

- создание рецепта;
- копирование рецептов других пользователей;
- возможность изменять скопированные рецепты под себя;
- отправка собственных рецептов модератору для публикации;
- планирование события с указанием рецептов;
- просмотр рецептов в офлайн режиме.

Модератор может выполнять эти же функции, а кроме этого, еще и давать возможность пользователям публиковать свои рецепты. Архитектурно проект является клиент-серверным приложением, где клиент с помощью интерфейса обеспечивает обмен информацией между пользователем и приложением, сервер принимает запросы клиента, обрабатывает их и возвращает ему результаты, а база данных отвечает за хранение информации от клиента. Серверная часть приложения написана на языке программирования JavaScript с применением программной платформы Node.js и фреймворка Express. Он имеет в своем распоряжении множество служебных методов HTTP и промежуточных обработчиков, которые позволяют создать надежный API быстро и легко.

Модель данных реализована с помощью СУБД MS SQL 2019, представляющей собой реляционную систему управления базами данных.

Для написания клиентской части кода использовалась среда разработки Microsoft VisualCode и AndroidStudio. Был применён фреймворк ReactNative. ReactNative – это кроссплатформенный фреймворк с открытым исходным кодом для разработки гибридных приложений на языке JavaScript с использованием TypeScript. Для эффективного менеджмента состояния клиентского приложения и сохранения данных в локальное хранилище использовалась библиотека Redux.