**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ**

1. **Разработка технического задания**
   1. Общие требования
   2. Назначение документа

Настоящее Техническое задание определяет требования и порядок разработки мобильного приложения для онлайн-магазина подарков Sunmul.

* 1. Наименование Исполнителя и Заказчика

Заказчик – онлайн-магазина Sunmul; Адрес: ул. Даргомыжского, 80, Нижний Новгород, Нижегородская обл., 603211;

Исполнитель – ИП GNA; Адрес: г. Н. Новгород, Московское ш., 1.

* 1. Информация о магазине

1. Название: онлайн-магазина Sunmul;
2. Слоган: «Подарок, преподнесенный с улыбкой, ценен вдвойне.»;
3. Отрасль и направления деятельности: розничная продажа товаров через Интернет (электронная коммерция);
4. Перечень услуг:
   * + Продажа подарков;
     + Сборка подарочных наборов;
     + Оформление подарочной упаковки;
     + Доставка.
5. Публичные контактные данные:
   * + - * Телефон: +79625062815
         * Электронная почта: 12bts.com@gmail.com;
6. Конкуренты: Зан–Зан, Люблю Дарить, Flowwow.
7. Основание для разработки приложения

Основанием для разработки приложения является Договор № 00000001 от 25.09.2024 между Исполнителем и Заказчиком.

* 1. Назначение приложения

Назначение мобильного приложения Sunmul заключается в предоставлении пользователям удобного и быстрого доступа к выбору, покупке и доставке подарков через их мобильные устройства.

Актуальность разработки приложения:

* приложение предоставляет пользователям возможность легко и быстро выбрать и получить подарки прямо с телефона, что особенно очень важно в моменты срочных покупок;
* приложение поможет повысить лояльность клиентов и даёт конкурентное преимущество на рынке;
* приложение позволит экономить время клиентов с помощью фильтров, поиска и простого интерфейса.
  1. Целевая аудитория
  2. Пол: женский, мужской;
  3. Возраст: от 12 лет;
  4. Занятость: школьники, студенты, люди пенсионного возраста и пр.;
  5. Цели и задачи приложения
  6. Тип приложения: многостраничное мобильное приложение.
  7. Цели приложения:
  + Обеспечение удобного и быстрого доступа к товарам, что способствует увеличению объема продаж;
  + Реализация различных способов оплаты и оформления заказа, чтобы сэкономить время для завершения покупки;
  + Оптимизация процессов доставки, позволяющая пользователям отслеживать статусы своих заказов;
  + Возможность фильтрации между различными категориями подарков.
  1. Целевые действия пользователей приложения:
* Регистрация пользователей;
* Авторизация пользователей;
* Редактирование профиля пользователя;
* Просмотр товаров онлайн-магазина;
* Фильтрация товаров по категориям;
* Просмотр карточки товара;
* Просмотр отзывов и рейтинга товара;
* Редактирование количества товаров в корзине;
* Оформление заказа;
* Выбор способа оплаты и доставки при оформлении заказа;
* Просмотр истории заказов.
  1. Стадии разработки приложения

Таблица №1 – Стадии разработки проекта

|  |  |  |
| --- | --- | --- |
| Стадии | Этапы | Сроки |
| Определение требований | Выявление и документирование потребностей и ожиданий заказчика/пользователя. | 4 часа |
| Определение функциональных и нефункциональных требований к новому ПО. |
| Проектирование | Разработка макета приложения. Проектирование интерфейсов и взаимодействия между ними. | 6 часов |
| Проектирование модулей интерфейсов и взаимодействия между ними. |
| Реализация (кодирование) | Написание исходного кода программы и подключение базы данных. | 32 часа |
| Разработка базы данных. |
| Вёрстка приложения. |
| Тестирование | Проверка функциональности, производительности, безопасности и других характеристик. | 8 часов |
| Выявление и устранение ошибок и дефектов. |
| Обеспечение качества и соответствие требованиям. |
| Подготовка к защите проекта | Подготовка общего отчета и презентации | 2 часа |
| Защита проекта | Демонстрация готового продукта | 2 часа |

* 1. Ресурсы

1. Человеческие ресурсы: разработчик и пользователи;
2. Инфраструктурные ресурсы: база данных для организации;
3. Программные ресурсы: ОС Windows 10, инструмент разработки Android Studio c инструментарием пользовательского интерфейса Jetpack Compose, графический онлайн – редактор Figma, система управления версиями Git, система хранения проектов Gogs, облачная платформа системы управления базами данных Supabase (созданная на основе PostgreSQL);
4. Информационные ресурсы: документация по языку программирования Kotlin.
   1. Нефункциональные требования
   2. Требования к персоналу

Технических специалистов узкого профиля для поддержания приложения не требуется, необходимыми и достаточными навыками будет владение общими понятиями работы с сенсорным телефоном на уровне базового пользователя и выше.

* 1. Требования к разграничению доступа

В соответствии с правами доступа можно выделить 1 категорию. пользователей сайта: Покупатели – доступ ко всей общедоступной информации в приложении, просмотр, редактирование и удаление.

* 1. Предварительная структура приложения

Основная страница приложения должна представлять собой экран с основными разделами, на которое выполняется переход после авторизации пользователя в системе.

Должна быть предусмотрена возможность перехода на следующие страницы с главной:

* Профиль пользователя;
* Редактирование профиля;
* Заказы;
* Карточка товара;
* Отзывы;
* История заказов.
  1. Элементы макета приложения

Структура страниц приложения должна состоять из следующих элементов:

* «Навигационные элементы» – горизонтальный блок снизу – в данном блоке должны располагаться ссылки на соответствующие функциональные страницы, на которые можно перейти;
* «Основная часть» – часть страницы, на которой представлен функционал/информация соответствующие данному блоку.
  1. Требования к оформлению и дизайну
  2. Логотип: должен быть отображен;
  3. Предпочтительные цвета: оттенки синего, светло-зелёного, белоснежный и светло-синий;
  4. Возможные предпочтительные шрифты: Jersey 10, Montserrat Alternates и дополнительные.
  5. Количество вариантов дизайна

Основная тема приложения – светлая, дополнительно по требованию Заказчика может быть разработана темная тема.

* 1. Информационное наполнение приложения

Требуется. Представление информации, хранящейся в базе данных.

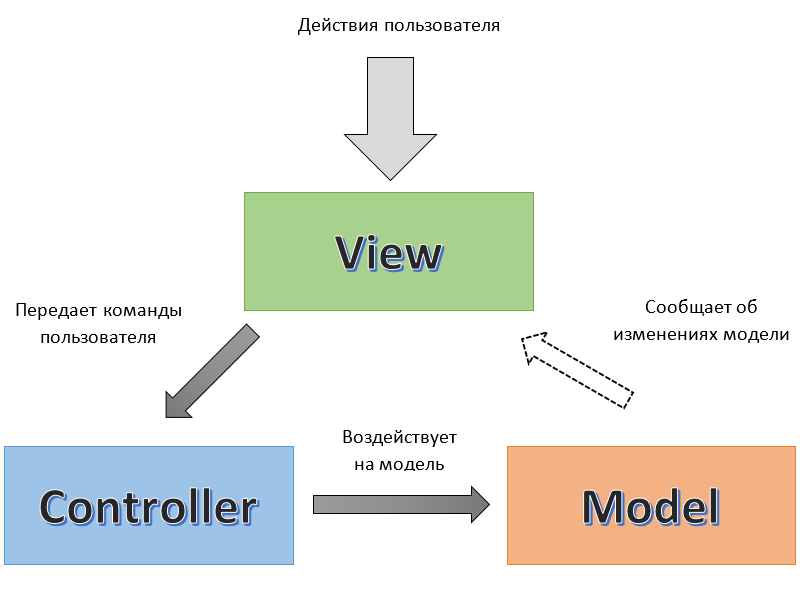
* 1. Функциональные требования
  2. Примерный набор модулей
  3. Регистрация и аутентификация пользователей;
  4. Редактирование корзины;
  5. Составление заказов и истории заказов;
  6. Вывод и создание отзывов пользователей;
  7. Управление личным кабинетом пользователя.
  8. Дополнительно
  9. Дополнительные пожелания

Дополнительные пожелания диктуются Заказчиком в процессе разработки;

1. **Выбора паттерна проектирования**

Первый паттерн, который будет рассмотрен это MVС. MVC – это шаблон программирования, который позволяет разделить логику приложения на три части:

* Model (модель). Получает данные от контроллера, выполняет необходимые операции и передаёт их в вид.
* View (вид или представление). Получает данные от модели и выводит их для пользователя.
* Controller (контроллер). Обрабатывает действия пользователя, проверяет полученные данные и передаёт их модели.

****Схема взаимодействия слоев в MVС представлена на рисунке №1.

Рисунок№1 – Схема взаимодействия слоёв в MVС

Преимущества:

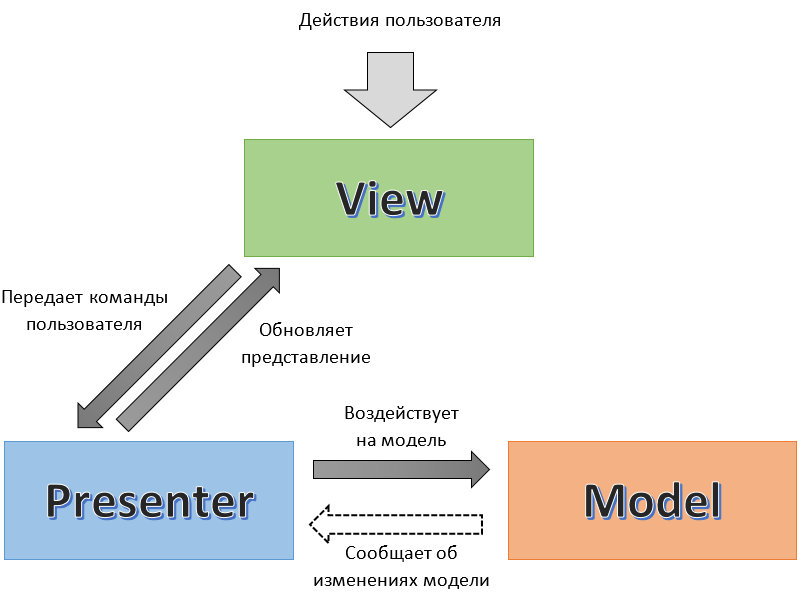
* разделяет бизнес–логику в модели;
* поддерживает асинхронные методы;
* модификация не затрагивает всю модель;
* более быстрый процесс разработки.

Недостатки:

* из–за большие количества кода в контроллере он может быть захламлен;
* мешает модульному тестированию.

Следующий паттерн, MVP. Это паттерн программирования графических интерфейсов. В нём приложение делится на три компонента:

* Model (Модель) работает с данными, проводит вычисления и руководит всеми бизнес–процессами.
* View (Вид или представление) показывает пользователю интерфейс и данные из модели.
* Presenter (Представитель) служит прослойкой между моделью и видом.

****Схема взаимодействия слоёв в MVP представлена на рисунке №2.

Рисунок№2 – Схема взаимодействия слоёв в MVP

Преимущества:

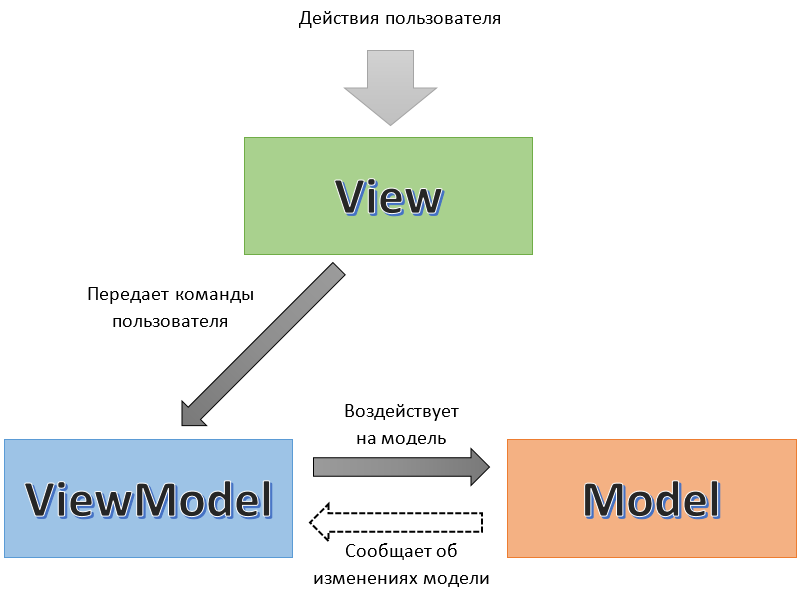
* делает View абстрактной, так что бы ее можно было легко поменять;
* переиспользование View и Presenter;
* код более читабельный и удобный в сопровождении;
* простое тестирование, так как бизнес–логика отделена от пользовательского интерфейса.

Недостатки:

* тесная связь между View и Presenter;
* огромное количество интерфейсов для взаимодействия между слоями;
* размер кода довольно избыточен.

Последним из рассмотренных паттернов будет MVVM – это паттерн разработки, позволяющий разделить приложение на три функциональные части:

* Model – основная логика программы (работа с данными, вычисления, запросы и так далее).
* View – вид или представление (пользовательский интерфейс).
* ViewModel – модель представления, которая служит прослойкой между View и Model.

****Схема взаимодействия слоёв в MVVM представлена на рисунке №3.

Рисунок№3 – Схема взаимодействия слоёв в MVVM

Преимущества:

* Разделение логики приложения и представления
* Гибкость и переиспользование
* Связывание данных
* Упрощение тестирования

Недостатки:

* Увеличение объема кода

После рассмотрения и анализа всех вышеперечисленных паттернов, заключительный выбор был сделать в пользу паттерна MVVM. Выбор пал именно на паттерн MVVM потому что, во-первых, хоть MVC и является классическим паттерном, который подходит для небольших проектов и прототипов, однако, по мере роста приложения, код контроллера может стать захламлённым, что затруднит поддержку и тестирование.

Во-вторых, MVP может и предлагает лучшее разделение логики, что упрощает тестирование и делает код более управляемым, но он требует больше усилий для создания дополнительных интерфейсов, что может усложнить архитектуру.

В итоге, паттерн MVVM, рекомендованный Google для разработки Android-приложений, предлагает наилучшие возможности для двусторонней привязки данных, модульного тестирования и разделения логики. Этот паттерн идеально подходит для крупных и сложных приложений, где требуется высокая гибкость и поддерживаемость.

1. **Проектирование требований к серверной части приложения**
2. Предметная область

Онлайн-магазин Sunmul занимается продажами готовых подарков и подарочных наборов.

База данных магазина хранит в себе список его клиентов, товаров, представленных в приложении, и их категорий, список заказов и историю покупок, а также отзывы на товары и корзину с товарами.

Каждый пользователь предоставляет приложению своё имя, электронную почту, пароль, телефон и день рождения.

О товаре пользователь может узнать его название, описание, цену, из какой он категории, статус, рейтинг, вес и страну.

В заказе пользователя отражается его дата оформления, статус, общая сумма, адрес и способ доставки, оплаты. В одном заказе может быть несколько товаров.

Отзывы содержат в себе оценку, комментарий и дату создания этого отзыва. В корзине пользователя храниться выбранный им товар и количество этого товара. История заказов содержит все совершённые покупки.

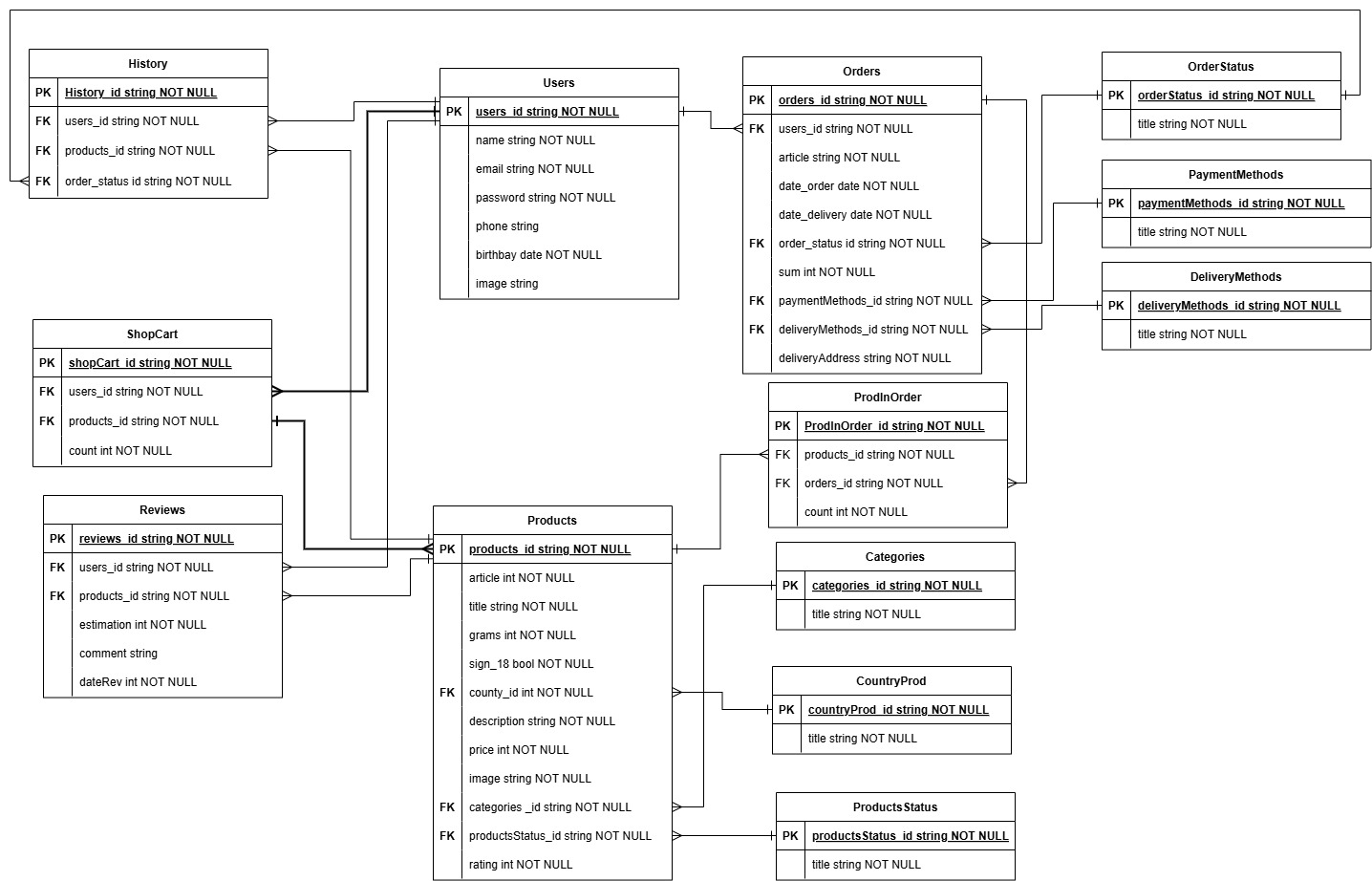
1. Описание основных сущностей предметной области

В результате проведенного анализа предметной области базы данных перечислим основные сущности (таблицы) базы данных.

В реляционную модель проектированной БД будут входить следующие таблицы (сущности): Пользователи (Users), Заказы (Orders), Товары (Products), Категории товаров (Categories), Товары в заказе (ProdInOrder), Отзывы (Reviews), Корзина (ShopCart), История товаров (History), Способы оплаты (PaymentMethods), Способы доставки (DeliveryMethods), Статус заказа (OrderStatus), Статус товара (ProdStatus), Страна производства (CountryProd).

Таблица №2 – Список сущностей

| № | Название | Назначение |
| --- | --- | --- |
| 1 | Пользователи | Регистрация/авторизация пользователя |
| 2 | Заказы | Создание списка заказов пользователя |
| 3 | Товары | Список товаров для пользователя |
| 4 | Категории товаров | Перечень категорий по которым можно фильтровать товар |
| 5 | Товары в заказе | Список товаров входящих в один заказ |
| 6 | Отзывы | Комментировать и выставлять оценку товарам пользователями |
| 7 | Статус заказа | Отображение статуса заказа для пользователя |
| 8 | Статус товара | Отображение статуса товара для пользователя |
| 9 | Корзина | Список товаров выбранных для покупки пользователем |
| 10 | История товаров | Список совершённых заказов |
| 11 | Способы оплаты | Перечень способов оплаты для выбора пользователем |
| 12 | Способы доставки | Перечень способов доставки для выбора пользователем |
| 13 | Страна производства | Список стран производства товара |

На рисунке №4 представлена ER–диаграмма базы данных для онлайн-магазина Sunmul.

Рисунок№4 – ER–диаграмма базы данных для онлайн-магазина Sunmul

Таблица №3 – Список связей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название связи | Сущности, участвующие в связи | Назначение |
| 1 | 1:М | Пользователи - Заказы | Один пользователь может сделать много заказов |
| 2 | 1:М | Пользователи - Отзывы | Один пользователь может написать много отзывов |
| 3 | 1:М | Товары - Отзывы | Один товар может иметь много отзывов |
| 4 | M:М | Пользователи - История,  История - Товары | Один пользователь может иметь много записей в истории, один товар может несколько раз повторяться в истории |
| 5 | 1:М | Пользователи - Корзина, | Один пользователь может иметь много товаров в корзине |
| 6 | 1:М | Корзина - Товары | В корзине может быть много товаров |
| 7 | 1:М | Статус заказа - Заказы | Один статус может быть у нескольких заказов |
| 8 | 1:М | Способы оплаты - Заказы | Один способ оплаты может быть у нескольких заказов |

Продолжение таблицы №3 – Список связей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название связи | Сущности, участвующие в связи | Назначение |
| 9 | 1:М | Способы доставки - Заказы | Один способ доставки может быть у нескольких заказов |
| 10 | 1: М | Заказы - Товары | Один заказ может иметь несколько товаров |
| 11 | 1: М | Товары в заказе - Товары | В заказе может быть много товаров с определённым количеством |
| 12 | 1: М | Категории - Товары | Одна категория может быть у нескольких товаров |
| 13 | 1: М | Статус - Товары | Один статус может быть у нескольких товаров |
| 14 | 1: М | Страна - Товары | Одна страна производства может быть у нескольких товаров |

Также потребуется интеграция Яндекс.Карт в мобильное приложение онлайн-магазина подарков, что позволит пользователям легко находить магазины или пункты самовывоза. Интеграция Яндекс.Карт сделает мобильное приложение удобным для пользователей, упростив им поиск ближайших точек продажи, а также взаимодействие с сервисом доставки.

1. **Проектирование требований к мобильному приложению**

Мобильное приложение Sunmul на старте предлагает пользователю зарегистрироваться в приложении с помощью ввода электронной почты и пароля. Вход в аккаунт в приложении будет осуществляться с помощью ввода логина и пароля. Пользователю будет предоставлен список товаров, который он может фильтровать по категориям. Каждый товар имеет подробную информацию с изображениями, описанием и характеристиками. Выбранный товар пользователь сможет отправить в корзину для последующего оформления покупки.

Также пользователь может редактировать корзину, изменять количество и удалять товары из корзины. Лёгкий и быстрый процесс оформления заказа с возможностью выбора способа оплаты и доставки. Возможность отслеживания статуса заказа (обрабатывается, отправлен, доставлен). При просмотре товара пользователь имеет доступ к отзывам других пользователей для более информированного выбора товаров и возможность оставлять свои отзывы о товарах и оценивать их. У пользователя будет возможность редактировать личные данные, адреса доставки и просматривать историю заказов в своём профиле.

Требования к мобильному приложению:

* Доступ к интернету;
* Обязательная регистрация;
* Версия операционной системы Android не ниже 5.0 (API level 21);
* Разрешение на доступ к геолокации обязательно, а к галерее и уведомлениям по желанию пользователя;
* Предоставление магазинов на карте, в которых можно приобрести выбранный товар в приложении.

1. **Выбор для реализации ПО и обоснование выбора**

Платформы для разработки приложений для Android упрощают процесс разработки за счет сокращения времени и затрат, поэтому выбор подходящего программного обеспечения очень важен. Рассмотрим три самые популярные платформы для разработки мобильного приложения.

1. React Native - это кроссплатформенная платформа для разработки приложений с открытым исходным кодом, разработанная Meta, которая может помочь разработчикам в создании собственных приложений для Android, iOS и Интернета. Для разработки, тестирования и развертывания приложения используются JavaScript и библиотека React. Это позволяет быстро разрабатывать и развертывать мобильные приложения.

Синтаксис React Native такой же, как у CSS, но он не использует HTML или CSS. Чтобы воспользоваться преимуществами встроенных представлений, используются сообщения потока JavaScript. Такие приложения, как Skype, Airbnb, Amazon Prime и др., разрабатываются с использованием React Native.

Характеристики:

* Приложение создано с использованием собственных элементов и API, которые обеспечивают удобство работы с пользователем.
* Компоненты можно использовать повторно, а расширения сторонних производителей можно легко интегрировать.
* С помощью React Native для интерфейсных приложений можно создать компонентный графический интерфейс для интерфейсных приложений.
* Единую кодовую базу можно использовать на всех платформах для создания платформенно-зависимых версий многочисленных функциональных возможностей.
* Предоставляет множество сервисов для создания серверной части приложения.

1. Flutter - кроссплатформенная платформа для разработки приложений для Android с открытым исходным кодом. Она была создана Google и позволяет разработчикам разрабатывать приложения для Android, Windows, iOS и Интернета с помощью единой кодовой базы. Она написана на языке Dart. Он содержит виджеты, инструменты отладки, движок рендеринга и инструменты для интеграции API для создания приложений. Такие приложения, как Google Ads, Google Pay, eBay Motors и т.д., разрабатываются с использованием Flutter.

Характеристики:

* В Flutter есть функция горячей перезагрузки, благодаря которой загружаются только изменения, а не все приложение целиком. Это очень полезно для экономии времени, поскольку результат можно увидеть в режиме реального времени с меньшим временем компиляции.
* В приложение могут быть интегрированы различные серверные функции и сервисы, такие как аутентификация пользователя (через Google, Facebook, электронную почту/пароль, телефон, Microsoft, GitHub и т.д.), облачные сервисы, хранилище и push-уведомления.
* Предоставляет различные анимации и функции улучшения пользовательского интерфейса, чтобы сделать приложение более привлекательным.
* Обеспечивает производительность, близкую к производительности Native.

1. Android Studio - официальная интегрированная среда разработки для Android от Google (IDE), которая предоставляет множество функций для разработки мобильных приложений. Разработчики могут легко писать исходный код, отлаживать и тестировать качество своего приложения. Поддерживает различные языки программирования, такие как Java, Kotlin, Swift, C ++ и др. Google постоянно обновляет его, чтобы соответствовать последним тенденциям и требованиям разработчиков, и позволяет им настраивать его в соответствии с вашими потребностями. У него большое сообщество разработчиков Android, которые помогут вам, если вы застряли.

Характеристики:

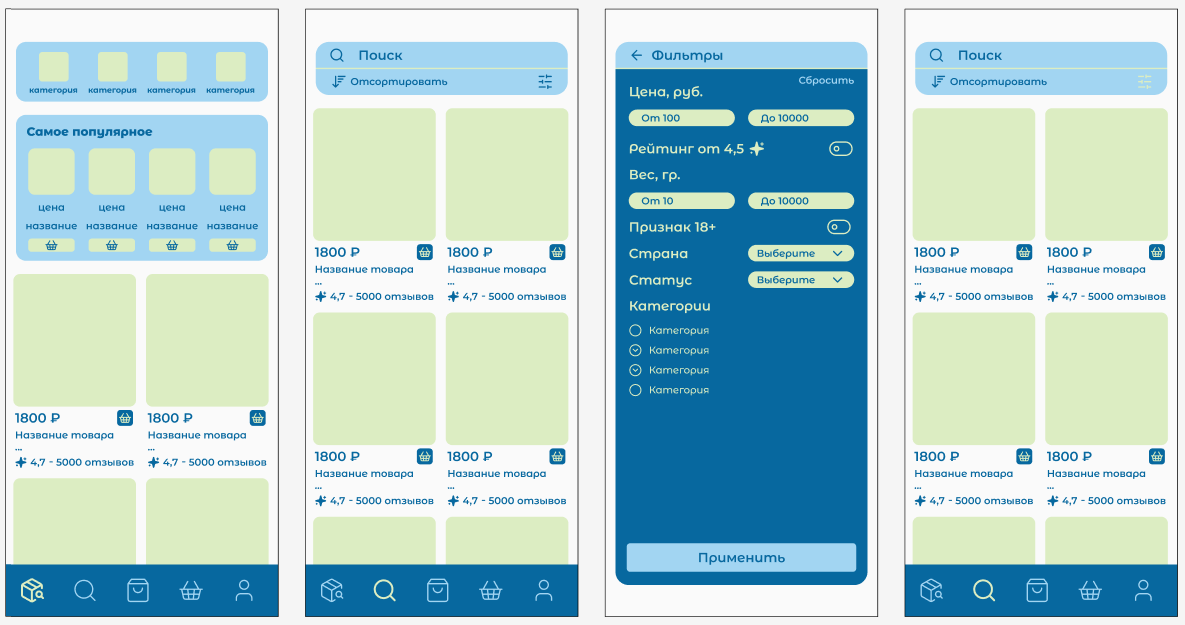
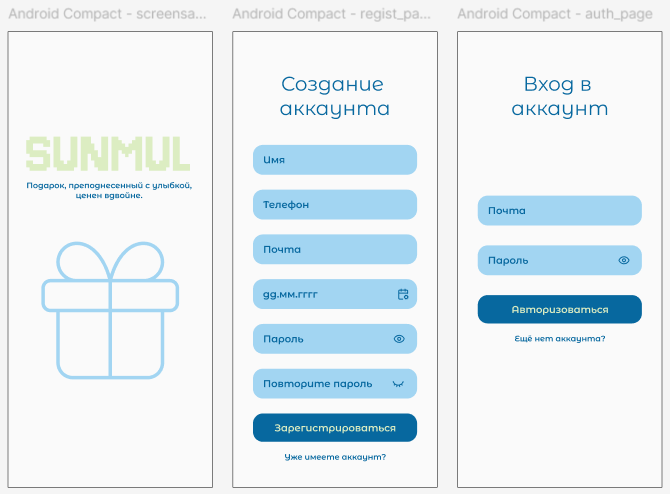
* Имеет виртуальные эмуляторы, а также физическое подключение к устройствам, как проводным, так и беспроводным.
* Поддерживает сборку Gradle
* Предоставляет сервисы Google Play, такие как Google Реклама, Google Карты, покупки в приложении и т.д.
* Поддерживает GitHub, поэтому можно выполнить любую команду Git всего несколькими щелчками мыши.
* Интегрирован с инструментами и фреймворками для тестирования.

Среди всех трёх платформ лучшего для проекта подойдёт Android Studio по следующим причинам:

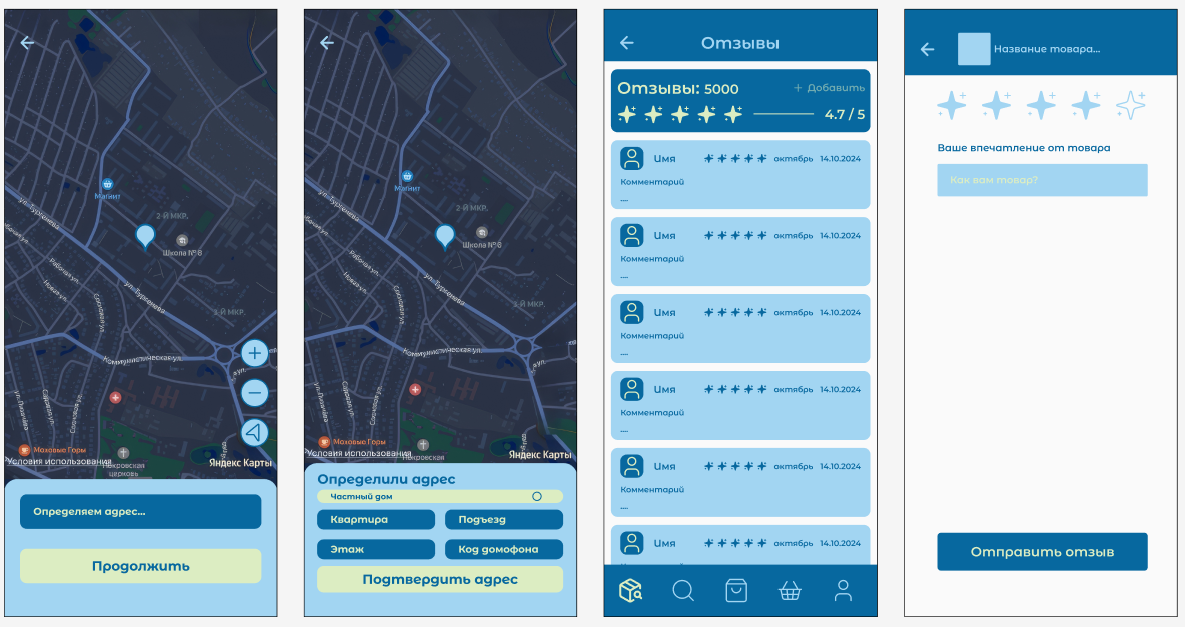
* Android Studio является официальной средой разработки для платформы Android, что означает, что все новые функции и обновления имплементируются именно в этой среде. Вы получите доступ к самым последним инструментам и API от Google.
* Kotlin — современный язык программирования, который идеально подходит для разработки приложений на Android. Он более лаконичный и безопасный, чем Java, и обеспечит разработку более качественного и чистого кода.
* Android Studio легко интегрируется с Supabase, облачной платформой от Google, предлагающей множество услуг, таких как аналитика, аутентификация пользователей, работа с базами данных в реальном времени и другие удобные инструменты.
* Встроенный эмулятор позволяет тестировать приложения на различных устройствах и версиях Android без необходимости физически иметь каждый из них.
* Android Studio поддерживает инструменты для работы с многопоточностью, что очень важно для создания отзывчивого и производительного приложения.

1. **Разработка макета приложения**

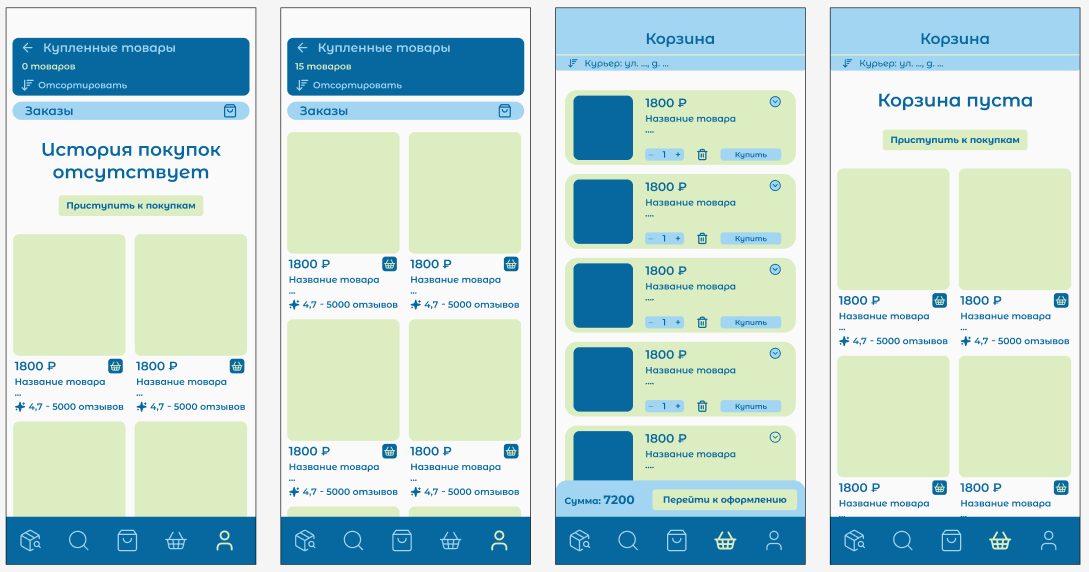
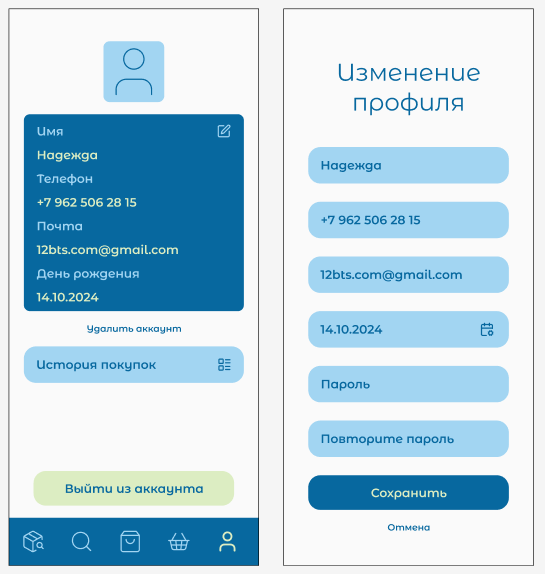
В условиях современного цифрового рынка, успешная реализация онлайн-магазина требует создания интуитивно понятного и визуально привлекательного интерфейса. Исходя из этого был разработан макет для приложения онлайн-магазина Sunmul. При разработке макета было создано следующее: страница с логотипом онлайн-магазина, страницы авторизации и регистрации, главная страница с товарами, страница для поиска и фильтрации товара, страница с карточкой товара, страницы заказов и их оформления, страницы с профилем пользователя и его редактирование, также были разработаны страницы с историей покупок и корзиной, страницы для просмотра и добавления отзывов, и страницы для определения адреса доставки на Яндекс картах. Ниже представлены рисунки (№5 – №10) со всеми страницами макета приложения.

Рисунок№5 – Страницы с логотипом, авторизацией и регистрацией

Рисунок№6 – Страницы с товарами, поиском и фильтрацией

Рисунок№7 – Страницы с карточкой товара, заказами и их оформлением

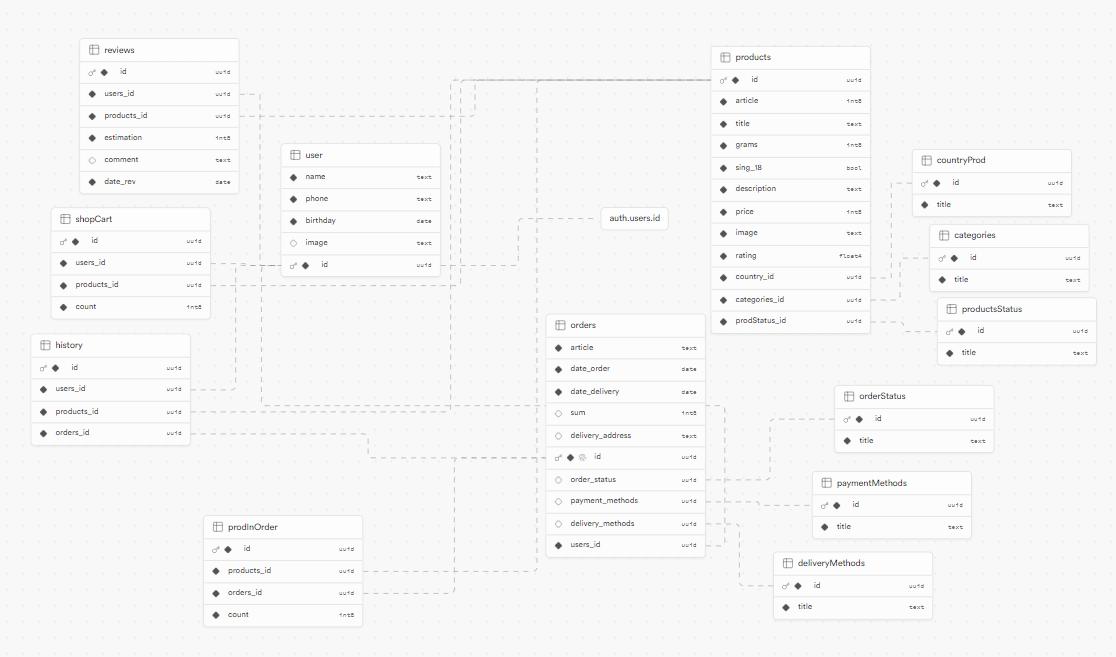
Рисунок№8 – Страницы с Яндекс картами, отзывами и их добавлением

Рисунок№9 – Страницы с корзиной и историей покупок

Рисунок№10 – Страницы с профиля и его редактирования

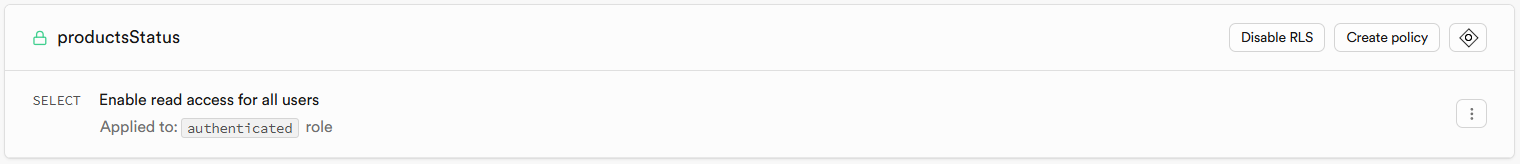
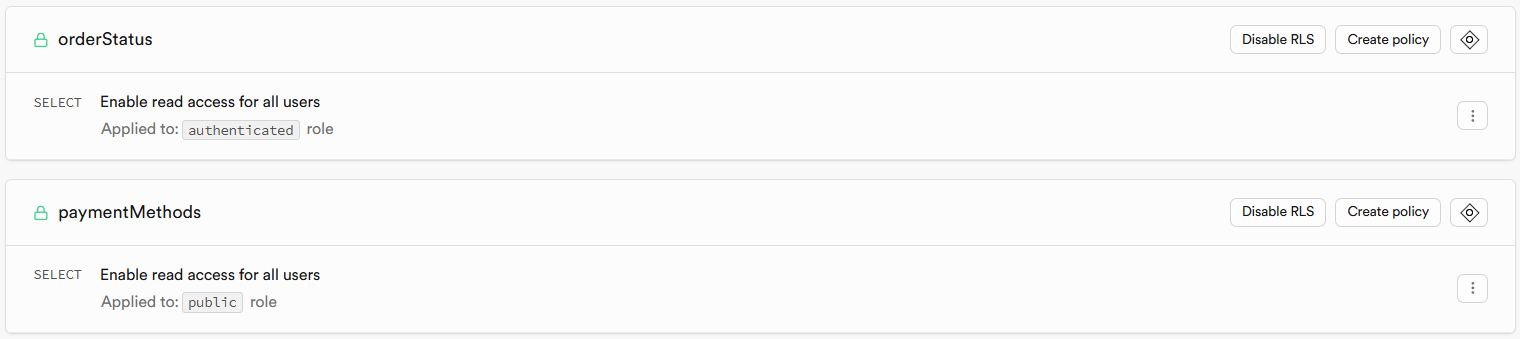
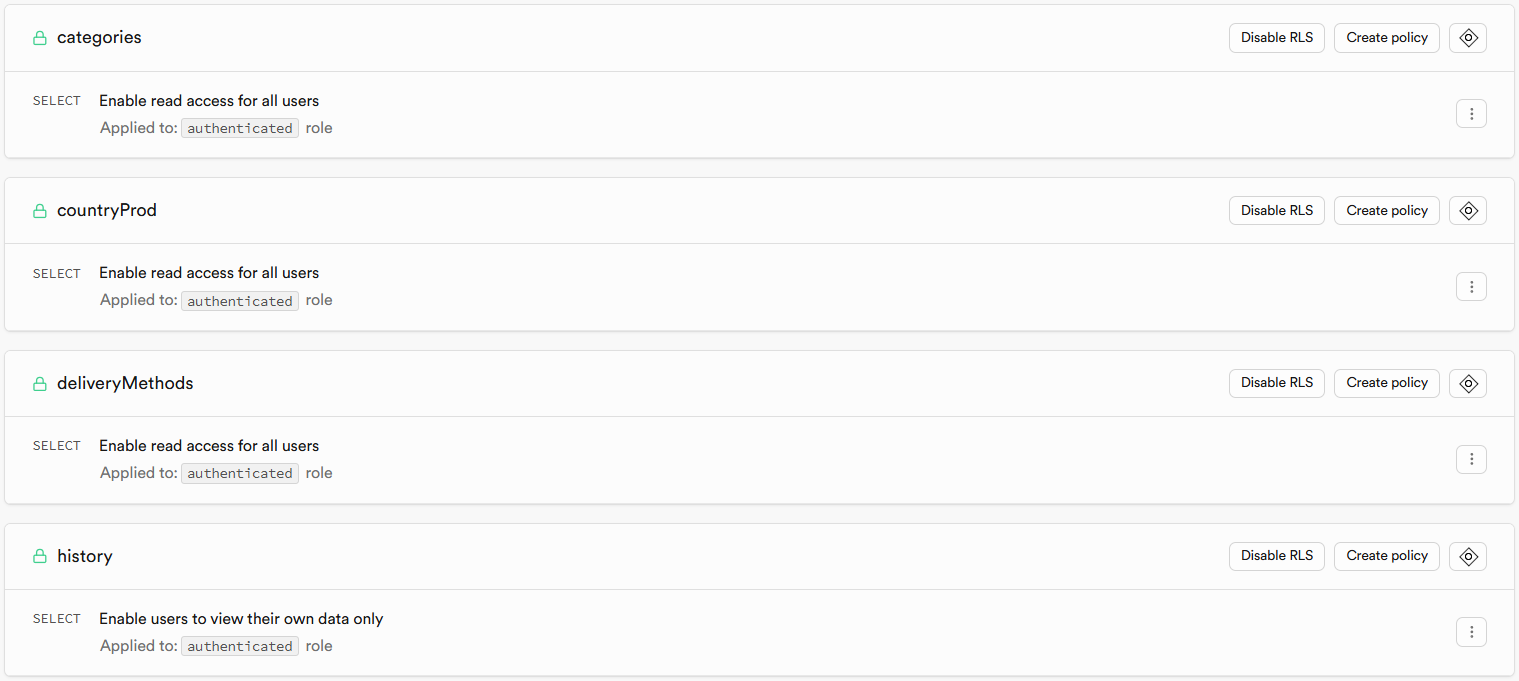
1. **Разработка серверной части приложения**

В качестве хранилища данных будет использоваться облачная платформа Supabase. Supabase — это мощная облачная платформа с открытым исходным кодом, предназначенная для разработчиков. Она предоставляет удобные инструменты для создания и управления базами данных, а также включает в себя аутентификацию пользователей, хранение данных и функции реального времени. Supabase построена на базе PostgreSQL, что обеспечивает надежность и масштабируемость. Платформа идеально подходит для быстрого прототипирования и разработки приложений, благодаря простоте использования и интеграции с популярными фреймворками.

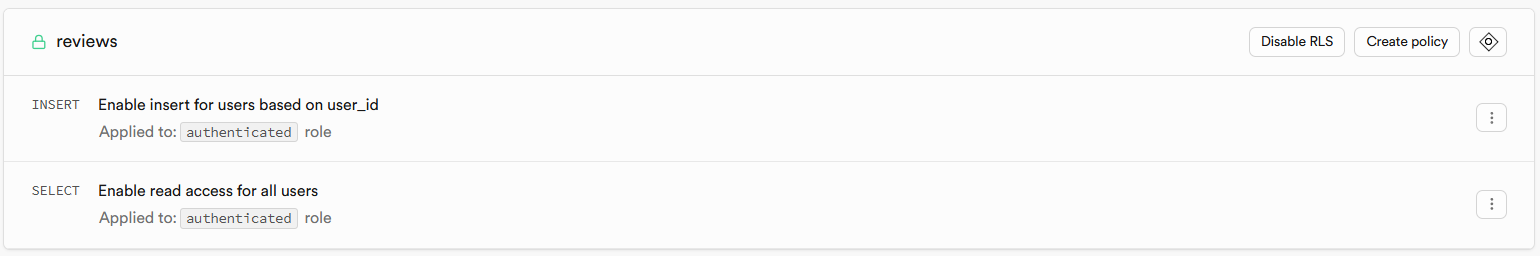
На данной платформе были созданы таблицы, которые необходимы для хранения нужных для приложения онлайн-магазина Sunmul данные. Ниже на рисунке №11 представлена схема, на которой отображены все созданные таблицы и какие данные содержит каждая таблица.

Рисунок№11 – Схема отображения таблиц в Supabase

Также для каждой таблицы были созданы политики, например, таким таблицам как: categories (Категории), countryProd (Страны производителей), deliveryMethods (Способы доставки), orderStatus (Статус заказа), paymentMethods (Способы оплаты), productsStatus (Статус товара), была создана политика на чтение данных из таблиц. На рисунке №12 представлено готовое создание политик для выше описанных таблиц.

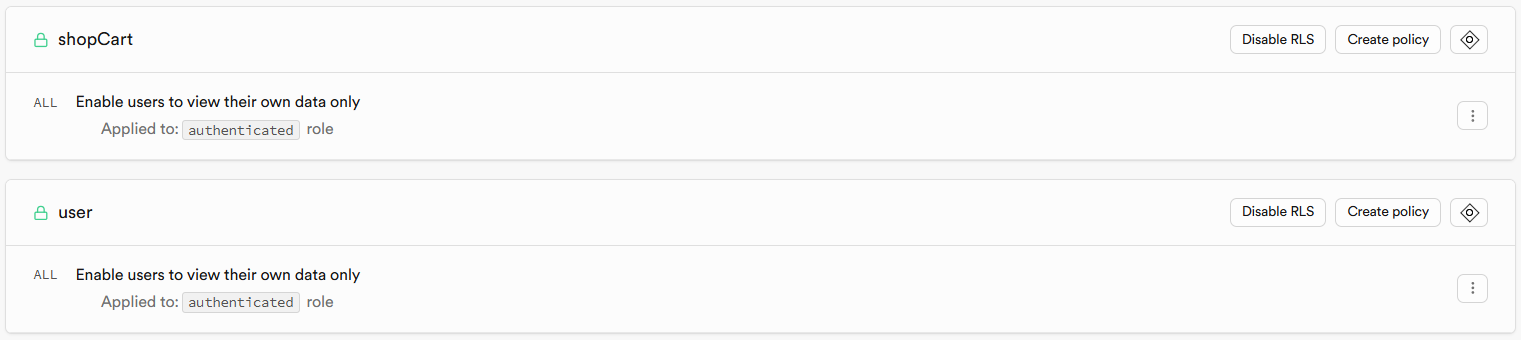
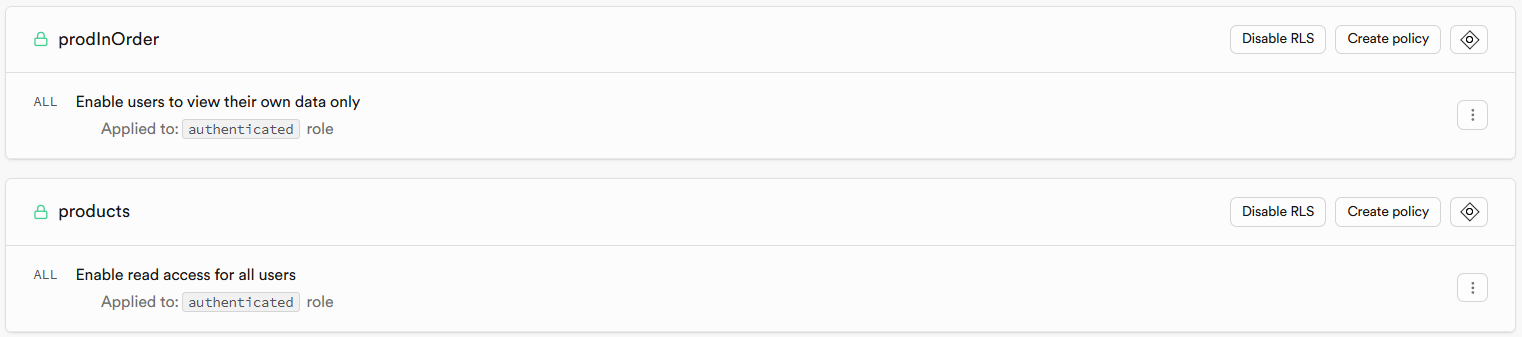
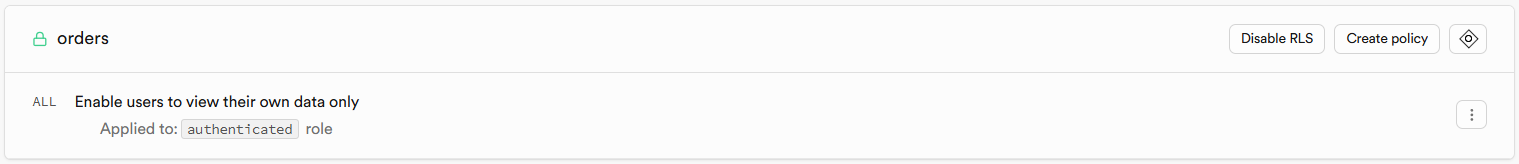


Рисунок№12 – Политики на чтение данных

Для таблицы reviews (Отзывы) также была создана политика на чтение, но и добавлена ещё политика на вставку данных в таблицу. Сделано это для того, чтобы пользователь мог оставлять свои отзывы на товар, и они бы хранились в данной таблице. На рисунке №13 представлено готовое создание политик для выше описанной таблицы.

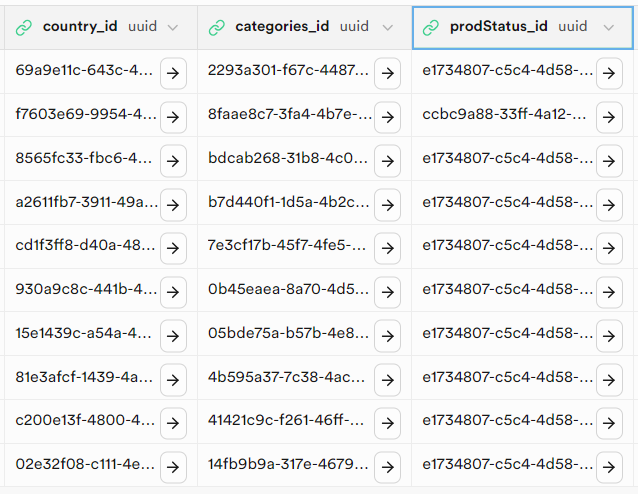
Рисунок№13 – Политики на чтение и вставку данных

Для таблиц history (История покупок), orders (Заказы), prodInOrder (Товары в заказе), products (Товары), shopCart (Корзина) и user (Пользователь) создана одна политика, которая позволяет читать, вставлять, обновлять и удалять данные. Это может быть полезно для обновления заказов, корзины и истории покупок, редактирования/удаления профиля пользователя и обновления рейтинга товара. На рисунке №14 представлено готовое создание политик для выше описанных таблицы.



Рисунок№14 – Политики на чтение, вставку, обновление и удаление данных

В качестве примера заполнения таблиц ниже представлен рисунок №15 с заполненной таблицей products (Товары). Аналогичным образом были заполнены и остальные таблицы базы данных.



Рисунок№15 – Заполненная таблица products (Товары)

1. **Тестирование серверной части приложения**
2. Тест-план
   1. Перечень работ

Предметом для тестирования является база данных онлайн-магазина Sunmul. Тестируемой частью будет безопасность данных хранящихся в таблицах БД при обращении к ним в запросах.

* 1. Критерии качества
* Надёжность: база данных должна быть надёжной и стабильной, с минимальной вероятностью сбоев и ошибок;
* Безопасность: база данных должна быть защищена от несанкционированного доступа, а также предусмотрены механизмы аутентификации и авторизации пользователей;
* Эффективность: производительность базы данных должна быть достаточной для обработки больших объёмов данных, что позволит быстро выполнять запросы и обновлять информацию;
* Функциональность: базы данных должна удовлетворять всем требованиям для функционирования онлайн-магазина, например, сохранять историю покупок;
* Масштабируемость: база данных должна быть масштабируемой, то есть позволять быстро и легко увеличивать её объём и мощность при необходимости.
  1. Оценка рисков

Оценка возможных рисков представлена в таблице №4. В качестве метрик для оценки в поле «Вероятность появления» используются варианты «редко/иногда/часто», а в поле «Критичность» – «маленькая/средняя/серьёзная».

Таблица №4 – Оценка и анализ рисков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Риск | Описание | Вероятность появления | Критичность | Варианты решения |
| Утечка данных клиентов | Нарушение безопасности, утечка личных данных клиентов (имена, адреса, пароли) | Часто | Серьёзная | Шифрование данных, двухфакторная аутентификация, регулярные тесты на уязвимости, контроль прав доступа |
| Неавторизованный доступ | Взлом базы данных, получение доступа к закрытым данным | Часто | Серьёзная | Ограничение прав доступа |
| Медленная работа базы данных | Долгое время отклика из-за высокой нагрузки или недостаточной оптимизации | Иногда | Средняя | Оптимизация запросов |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сбой при обновлении базы данных | Проблемы при обновлениях, приводящие к сбоям в работе или потере данных | Иногда | Серьёзная | Тестирование обновлений |
| Ошибки в данных | Ошибочные или устаревшие данные о товарах или клиентах | Редко | Средняя | Автоматическая валидация данных, регулярные проверки данных на целостность |
| Отказ оборудования | Выход из строя сервера, где расположена база данных | Иногда | Серьёзная | Регулярное тестирование отказоустойчивости системы |

Продолжение таблицы №4 – Оценка и анализ рисков

* 1. Документация:

1. Тест-план;
2. Стратегия тестирования;
3. Ресурсы;
4. Тестовый набор;
5. Тест-кейс.
   1. Стратегия тестирования

Таблица №5 – Стратегия тестирования

|  |  |
| --- | --- |
| Вид теста | Тесты |
| Функциональное тестирование | Возможность получения данных из БД |
| Возможность вставки данных из БД |
| Возможность обновления данных из БД |
| Возможность удаления данных из БД |
| Возможность получения данных из БД по запросу с фильтрацией |
| Бета-тестирование | Предоставить возможность доступа к БД группе пользователей для сбора обратной связи по удобству использования |
| Верификация | Проверка соответствия значений в БД требованиям (формат телефона, даты) |
| Тестирование безопасности | Проверка шифрования персональных данных |
| Проверка на ограничение доступа к данным |

Продолжение таблицы №5 – Стратегия тестирования

|  |  |
| --- | --- |
| Стресс тестирование | Возможность БД обрабатывать высокую нагрузки при большом количестве запросов |
| Возможность БД стабильно работать при большом объёме данных |
| Тестирование документации | Полнота и актуальность документации по использованию БД и её структуре |
| Валидация | Возможность обрабатывать некорректные данные в запросах к БД |
| Тестирование входных точек | Тестирование начальной загрузки данных в БД |
| Юзабилити-тестирование | Проверка удобства использования БД в лёгкости понимания её структуры и нахождения необходимых данных |
| Компонентное тестирование | Проверка корректности сущностей в БД |
| Проверка корректности связей между сущностями в БД |
| Интеграционное тестирование | Проверка корректности каскадного удаления данных из БД |
| Проверка работоспособности БД со сторонней системой Яндекс.Карт |

* 1. Ресурсы
* программные: браузер, СУБД Supabase созданная на основе PostgreSQL;
* аппаратные: компьютер, способный хранить и реализовывать потенциал операционной системы, текстовый редактор;
* временные: около 4-х часов;
* человеческие: один человек способный разрабатывать базы данных.

1. Тестовый набор

Таблица №6 – Тест-кейс №1

|  |  |
| --- | --- |
| ТС0001 Создание пользователя в таблице Auth в Supabase | |
| Предусловие: создана организация в Supabase | |
| Перейти на вкладку Authentication | Отобразить таблицу, в которой будут храниться зарегистрированные пользователи |
| По кнопке «Add user» добавить нового пользователя | Система Supabase открывает форму для заполнения почты и пароля для создаваемого пользователя |
| Ввести все необходимые данные и нажать на кнопку «Создать пользователя» | Система Supabase успешно создала нового пользователя и отобразила его в таблице Auth |

Таблица №7 – Тест-кейс №2

|  |  |
| --- | --- |
| ТС0002 Удаление пользователя в таблице Auth в Supabase | |
| Предусловие: создана организация в Supabase | |
| Перейти на вкладку Authentication | Отобразить таблицу, в которой будут храниться зарегистрированные пользователи |
| Нажать на строку с нужным пользователем, которого нужно удалить | Система Supabase открывает форму с информацией и настройками для выбранного пользователя |
| Нажать на кнопку «Delete user» в пункте «Danger zone» | Система Supabase выведет предупреждающее сообщение об удалении пользователя и всех связанных с ним данных |
| Нажать на кнопку «Delete user» | Система Supabase удалит выбранного пользователя из таблицы Auth |

Таблица №8 – Тест-кейс №3

|  |  |
| --- | --- |
| ТС0003 Получение данных из таблиц в Supabase | |
| Предусловие: создана организация в Supabase и созданы необходимые таблицы для БД | |
| Перейти на вкладку SQL Editor | Отобразить страницу, на которой можно прописывать SQL-запросы |
| Для вывода данных из нужных таблиц, созданные в БД, написать запрос, например, «select \*  from orders» и нажать на сочетание клавиш «CTRL» и «Enter», или на кнопку «Run» | Система Supabase отобразит данные, которые хранятся в запрашиваемой таблице, в зависимости от роли пользователя (администратор БД, анонимный пользователь, авторизированный пользователь).  Анонимному пользователю никакие данные не выведутся.  Администратору БД все записи выведутся.  Авторизированному пользователю выводятся, только те записи которые привязаны к его id. |

Таблица №9 – Тест-кейс №4

|  |  |
| --- | --- |
| ТС0004 Обновление данных в таблицах в Supabase | |
| Предусловие: создана организация в Supabase и созданы необходимые таблицы для БД | |
| Перейти на вкладку SQL Editor | Отобразить страницу, на которой можно прописывать SQL-запросы |
| Для обновления данных в нужных таблицах, созданные в БД, написать запрос, например, «update orders set delivery\_address = 'Нижний Новгород, 2-й микрорайон,д. 7,кв. 48, под. 5, дмф. 48К, этаж 1'  where article = '123456789-0010';» и нажать на сочетание клавиш «CTRL» и «Enter», или на кнопку «Run» | Система Supabase обновит данные, указанные в запросе, в зависимости от роли пользователя (администратор БД, анонимный пользователь, авторизированный пользователь).  Анонимному пользователю никакие данные не получиться изменить.  Администратор БД успешно изменит данные.  Авторизированному пользователю получится изменить данные, но только те записи которые привязаны к его id. |

Таблица №10 – Тест-кейс №5

|  |  |
| --- | --- |
| ТС0005 Добавление данных в таблицы в Supabase | |
| Предусловие: создана организация в Supabase и созданы необходимые таблицы для БД | |
| Перейти на вкладку SQL Editor | Отобразить страницу, на которой можно прописывать SQL-запросы |
| Для добавления данных в нужные таблицы, созданные в БД, написать запрос, например, «insert into reviews (users\_id, products\_id, estimation, comment, date\_rev) values ('ef2fa564-e3be-4b4f-a9cb-975c9bcc7f54','ba8d4705-426c-47c4-a859-54d94bffd5c1',5,'Недавно приобрела набор для гриля "Master Grilling Set" и осталась в полном восторге! Этот набор просто незаменим для любителей готовить на свежем воздухе. Качество материалов на высшем уровне: все инструменты очень прочные и удобные в использовании. Особо порадовали щипцы и лопатка – они отлично сбалансированы и позволяют легко переворачивать блюда.','2024-10-30');» и нажать на сочетание клавиш «CTRL» и «Enter», или на кнопку «Run» | Система Supabase добавит новую запись в указанную таблицу, в зависимости от роли пользователя (администратор БД, анонимный пользователь, авторизированный пользователь). Анонимному пользователю никакие данные не получиться вставить. Администратор БД успешно вставит данные. Авторизированному пользователю получится вставить данные, но только те записи которые привязаны к его id. |

Таблица №11 – Тест-кейс №6

|  |  |
| --- | --- |
| ТС0006 Удаление данных в таблицы в Supabase | |
| Предусловие: создана организация в Supabase и созданы необходимые таблицы для БД | |
| Перейти на вкладку SQL Editor | Отобразить страницу, на которой можно прописывать SQL-запросы |
| Для удаления данных в нужных таблицах, созданные в БД, написать запрос, например, «delete from "shopCart" where products\_id in (select P.id from products as P where P.title = 'Йога-мат "Eco Yoga Mat"');» и нажать на сочетание клавиш «CTRL» и «Enter», или на кнопку «Run» | Система Supabase удалит существующую запись в указанной таблице, в зависимости от роли пользователя (администратор БД, анонимный пользователь, авторизированный пользователь). Анонимному пользователю никакие данные не получиться удалить. Администратор БД успешно удалит данные. Авторизированному пользователю получится удалить данные, но только те записи которые привязаны к его id. |

Таблица №12 – Тест-кейс №7

|  |  |
| --- | --- |
| ТС0007 Фильтрация данных в таблицы в Supabase | |
| Предусловие: создана организация в Supabase и созданы необходимые таблицы для БД | |
| Перейти на вкладку SQL Editor | Отобразить страницу, на которой можно прописывать SQL-запросы |
| Для фильтрации данных в нужных таблицах, созданные в БД, написать запрос, например, «select article, P.title, price, grams, description from products as P join categories as C on C.id = P.categories\_id where C.title = 'Книги'» и нажать на сочетание клавиш «CTRL» и «Enter», или на кнопку «Run» | Система Supabase отфильтрует данные в указанной таблице, в зависимости от роли пользователя (администратор БД, анонимный пользователь, авторизированный пользователь).  Анонимному пользователю никакие данные не выведутся.  Администратору БД успешно выведутся отфильтрованные данные.  Авторизированному пользователю успешно выведутся отфильтрованные данные. |

Таблица №13 – Тест-кейс №8

|  |  |
| --- | --- |
| ТС0008 Поиск данных в таблицы в Supabase | |
| Предусловие: создана организация в Supabase и созданы необходимые таблицы для БД | |
| Перейти на вкладку SQL Editor | Отобразить страницу, на которой можно прописывать SQL-запросы |
| Для поиска данных в нужных таблицах, созданные в БД, написать запрос, например,  «select \* from products where title like 'Набор%'» и нажать на сочетание клавиш «CTRL» и «Enter», или на кнопку «Run» | Система Supabase найдёт данные в указанной таблице, в зависимости от роли пользователя (администратор БД, анонимный пользователь, авторизированный пользователь).  Анонимному пользователю никакие данные не выведутся.  Администратору БД успешно выведутся найденные данные.  Авторизированному пользователю успешно выведутся найденные данные. |

Таблица №14 – Тест-кейс №9

|  |  |
| --- | --- |
| ТС0009 Регистрация пользователя в системе Supabase | |
| Предусловие: создана организация в Supabase | |
| Открыть форму регистрации в приложении онлайн-магазина. Ввести необходимые данные (почту, пароль, имя, телефон, день рождения). Нажать на кнопку «Зарегистрироваться» | Приложение считывает введённые данные, обрабатывает их на корректность и записывает их в базу данных Supabase, почту и пароль в таблицу Auth, а остальные данные в таблицу User |

Таблица № – Тест-кейс №10

|  |  |
| --- | --- |
| ТС0010 Авторизация пользователя в системе Supabase | |
| Предусловие: создана организация в Supabase | |
| Открыть форму авторизации в приложении онлайн-магазина. Ввести необходимые данные (почту, пароль). Нажать на кнопку «Авторизоваться» | Приложение считывает введённые данные, обрабатывает их на корректность и сравнивает их с данными что уже имеются в базе дынных Supabase, и если данные сходятся, то пользователь авторизуется и переходит на главную страницу приложения |