1. **Анализ задачи**
   1. **Постановка задачи**
      1. **Организационно-экономическая сущность задачи**

Наименование задачи: разработка сайта и приложения для магазина одежды «IvantexShop».

Цель разработки: создание удобного онлайн-магазина для продажи модной одежды с широким ассортиментом и удобным интерфейсом для пользователей.

Назначение: ПП предназначен для клиентов, желающих приобрести модную одежду, и администраторов магазина для работы с информацией о товарах и клиентской базой.

Периодичность использования: ежедневно, с возможностью посещения магазина в любое время суток.

Источники и способы получения данных: информация о магазине «IvantexShop», его ассортименте, акциях, услугах.

Информационная связь с другими задачами: проект тесно связан с задачами администрирования интернет-магазина, клиентской базой и системой обработки заказов.

Обзор существующих аналогичных ПП: анализ аналогичных платформ, таких как

<https://7745.by/catalog>, показывает востребованность удобного интерфейса с функцией поиска и фильтрации, отзывов, а также наличия дополнительных сервисов, таких как персонализированные рекомендации, программа лояльности и система акций. Эти решения помогают повысить удобство и вовлеченность пользователей, что также может быть полезно для разработки функционала «IvantexShop».

* + 1. **Функциональные требования**

Описание перечня функций и задач, которые должно выполнять будущее ПП:

Гость:

* просмотр каталога товаров и поиск с возможностью фильтрации: каждый товар будет иметь фото, названия, описание, цену, жанр, издательство. Поиск будет осуществляться в отдельной строке в шапке сайта или приложения. Фильтры будут по категориям: цена, жанр, издательство;
* просмотр местоположения и время работы магазина;
* просмотр скидок на товары;
* переход в иные соц. сети магазина;
* регистрация и аутентификация гостя: пользователь при регистрации вводит свое ФИО, номер телефона, адрес, данные карты, придумывает пароль, электронную почту. При аутентификации вводит свою почту и пароль.

Пользователь – все те же функции, что может выполнять гость, но также появляется:

* авторизация;
* покупка в интернете (приложение);
* бронирование товаров, для дальнейшей их покупки;
* добавить в корзину (приложение);
* удалить из корзины (приложение);
* изменить количество товаров в корзине (приложение);
* посмотреть итоговую сумму корзины (приложение);
* создание и редактирование профиля;
* выход из личного кабинета.

Администратор:

* управление заказами: просмотр количества заказов и их данных, отмена заказов при желании пользователя, редактирование статуса заказа;
* работа с акциями и скидками (добавление, удаление и редактирование);
* отправка уведомлений о состоянии заказа на электронную почту: будут приходить уведомления о том, что заказ собирается, отправлен и прибыл на место заказа.
  + 1. **Описание входной информации**
* данные о товарах (фото, названия, описание, цену, жанр, издательство)**;**
* данные о клиентах (ФИО, номер телефона, адрес, данные карты, пароль, электронная почта)**;**
* данные заказа (количество и др.)**.**
  + 1. **Описание выходной информации**
* подтверждение заказа, информация о статусе заказа, квитанция о покупке**;**
* формы представления: электронные письма с информацией о заказе;
* периодичность и сроки представления: моментально после оформления заказа.
  + 1. **Перечень условно-постоянной информации**
* категории товаров**;**
* цены.
  + 1. **Нефункциональные (эксплуатационные) требования**

Требования к применению:

Пользовательский интерфейс должен быть интуитивно понятным и привлекательным для пользователей всех возрастов.

Требования к производительности:

Время загрузки страницы не должно превышать 5 секунд.

Система должна поддерживать одновременную работу не менее 500 активных пользователей без значительного снижения производительности.

Время ответа сервера на запросы должно быть меньше 200 миллисекунд для обеспечения отзывчивости интерфейса.

Требования к реализации:

Для разработки следует использовать современные технологии для обеспечения эффективности и масштабируемости.

Все данные должны храниться в защищенной базе данных, с шифрованием конфиденциальной информации для обеспечения безопасности пользовательских данных.

Приложение должно быть развернуто на облачной инфраструктуре с высокой доступностью и автоматическим масштабированием для обеспечения непрерывной работы.

Требования к надежности:

Система должна быть способна обрабатывать сбои и восстанавливаться автоматически без потери данных.

Допустимое время простоя системы в случае сбоя не должно превышать 1 часа в месяц.

Регулярные резервные копии данных должны создаваться ежедневно и храниться в надежном и защищенном месте.

Требования к интерфейсу:

Система должна быть доступна через различные устройства, включая компьютеры, планшеты и мобильные телефоны.

Интерфейс должен быть совместим с различными браузерами, такими как Google Chrome, Mozilla Firefox и Microsoft Edge.

На главной странице будет возможность просмотра каталога товаров с фильтрацией, просмотр информации о местонахождении магазина и время его работы, просмотр соц. сетей магазина и ссылка на установку приложения.

* 1. **Диаграмма вариантов использования**

Диаграмма вариантов использования выглядит следующим образом (приложение А).

* 1. **Выбор стратегии разработки и модели жизненного цикла**

Для разработки приложения «Интернет магазина «IvantexShop»» следует выбрать стратегию разработки и модель жизненного цикла. Осуществляем выбор посредством составления таблиц:

Таблица 1 - Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да |  |  |  |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да |  |  |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет |  |  |
| 4. | Нужно ли демонстрировать требования с целью их определения? |  |  | Да |  | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или системы? | Нет | Нет |  | Нет |  |  |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? |  |  |  | Да | Да | Да |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? |  |  | Да | Да | Да | Да |

Таблица 2 **–** Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков  проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет |  |  |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? |  |  | Нет | Нет | Нет |  |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет |  |  |  |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? |  |  | Нет |  | Нет | Нет |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да |  |  |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да |  | Да | Да | Да |

Таблица 3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да |  | Да |  | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Таблица 3 продолжение | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет |  | Нет |  | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? |  |  |  |  | Да | Да |

Таблица 4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. | Разрабатыва ли в проекте продукт для организации направления? | Нет | Нет | Нет |  |  |  |
| 2. | Будет ли проект являться расширением существующей системы? |  |  |  |  | Нет | Нет |
| 3. | Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет |  |  |  |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Да |  | Да |  | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? |  | Да |  | Да |  | Да |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? |  |  |  | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет |  |  |  |
| 8. | Является ли график сжатым? | Нет | Нет |  |  |  |  |
| 9. | Предполагается ли повторное использование компонентов? |  |  | Да | Да | Да | Да |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? |  |  |  |  | Да | Да |

В итоге заполнения таблиц 1-4 наиболее подходящим жизненным циклом является V-образная модель

* 1. **Инструменты разработки**

Для разработки данного проекта будут использованы языки программирования такие как: Java и Kotlin, каждый из которых выполняет свою важную функцию в разработке веб-сайта.

Java — это один из основных языков программирования для Android-разработки. Причины выбора Java могут включать:

* **Широкая поддержка**: Java поддерживается практически на всех платформах, и Android — не исключение. Это один из официальных языков для разработки под Android, что обеспечивает ему отличную поддержку в SDK и документации.
* **Большое сообщество**: Огромное количество разработчиков используют Java, что упрощает поиск решений для возникающих проблем, множество готовых библиотек и инструментов, а также доступ к обучающим ресурсам.
* **Строгость типизации**: Java — строго типизированный язык, что позволяет снизить количество ошибок, связанных с неправильным использованием типов данных, и облегчает отладку кода.

Kotlin — это современный язык программирования, разработанный компанией JetBrains и ставший официальным языком для разработки под Android с 2017 года. Выбор Kotlin может быть обусловлен следующими причинами:

* **Современность и лаконичность**: Kotlin предлагает более лаконичный и современный синтаксис по сравнению с Java. Например, в Kotlin можно писать меньше кода для выполнения тех же задач, что делает его более удобным и менее подверженным ошибкам.
* **Совместимость с Java**: Kotlin полностью совместим с Java, что позволяет использовать существующие Java-библиотеки и интегрировать Kotlin-код в проекты на Java. Это также упрощает миграцию с Java на Kotlin.
  + - **Популярность и поддержка Google**: Kotlin активно поддерживается Google, и с каждым годом его использование среди Android-разработчиков растет. Большинство современных Android-приложений разрабатываются либо полностью на Kotlin, либо на комбинации Kotlin и Java.

Комбинированное использование Java и Kotlin

Использование обоих языков в одном проекте позволяет извлечь выгоду из преимуществ каждого из них. Например, вы можете использовать Java для стабильного и проверенного временем кода, а Kotlin — для новых функциональностей, где важны современность, лаконичность и безопасность. Это также позволяет постепенно мигрировать существующие приложения с Java на Kotlin без необходимости переписывать все с нуля.

1. **Проектирование задачи**
   1. **Разработка структуры сайта, системы меню, навигации**

Для разработки структуры сайта продажи товаров будет создана диаграмма, отражающая взаимосвязь между ключевыми страницами: "Главная", "Каталог", "Контакты". Будет разработана удобная система меню и навигации, обеспечивающая быстрый доступ к категориям, фильтрам, поиску и личному кабинету, что значительно улучшит пользовательский опыт и облегчит процесс покупок (приложение В).

* 1. **Разработка UML-диаграмм**

В рамках разработки проекта будут созданы UML-диаграммы, такие как диаграмма деятельности, диаграмма объектов и диаграмма классов. Диаграмма деятельности отразит последовательность действий пользователей на сайте, диаграмма объектов покажет структуру хранения данных, связи между таблицами и их атрибуты, а диаграмма классов покажет структуру системы, взаимодействие между классами, их атрибуты и методы, а также связи, такие как ассоциации, агрегации и наследование. Эти диаграммы помогут визуализировать архитектуру проекта и упростят процесс разработки (приложение Б).

* 1. **Разработка пользовательского интерфейса**

Пользовательский интерфейс был разработан в приложении «Figma» (приложение В).

Дизайн приложения выполнен в современном минималистичном стиле с использованием мягких цветовых переходов, ярких акцентов и четкой структуры. Основной фон имеет светло-розовый оттенок, создавая приятный и гармоничный визуальный эффект. В верхней части экрана расположен баннер с градиентом, плавно переходящим между фиолетовым, розовым и оранжевым цветами, на котором размещен логотип белого цвета и слоган приложения: «Магазин женской одежды. Легко, быстро, удобно!».

Карточки товаров представлены в адаптивной сетке с закругленными краями, каждая из которых включает изображение товара на нейтральном фоне, метки (например, красная метка «%» для скидок или сиреневая «new» для новинок), цену товара в отдельном розовом блоке с черным текстом и яркую кнопку «Купить» цвета фуксии (#FF00FF) с черным текстом и закругленными краями.

Навигационная панель расположена внизу экрана и выполнена в светло-розовом цвете, с минималистичными черными значками для перехода между разделами (домашняя страница, поиск, корзина, уведомления и профиль). Для шрифтов используется современный стиль без засечек, например, Arial, Roboto или Open Sans. Размер текста варьируется: заголовки крупные (24–28 пикселей), цена и описание товаров — около 16 пикселей, что обеспечивает удобство чтения.

Все элементы приложения организованы с равномерными отступами, а сетка адаптируется к размерам экрана, сохраняя удобство взаимодействия. Цветовая палитра, сочетающая светло-розовый фон с яркими акцентами, подчеркивает женственность и современность дизайна, создавая интуитивно понятный и эстетически привлекательный интерфейс.

1. **Реализация**
   1. **Создание сайта**

Создание приложения с помощью конструктора [GiveMeApp](https://gmaproject.ru) — это простой процесс, который не требует навыков программирования. Для начала необходимо зарегистрироваться на сайте, нажав на кнопку "Регистрация" и заполнив все необходимые поля. После подтверждения учетной записи вы можете войти в систему, используя свои данные.

В личном кабинете выберите опцию "Создать приложение", укажите название будущего приложения и выберите его категорию. На следующем этапе вы сможете выбрать один из стандартных шаблонов оформления, предложенных платформой. Дизайн можно персонализировать, изменив цветовую схему, шрифты и добавив собственные элементы.

После выбора шаблона начните настраивать структуру приложения. Вы можете добавить разделы, такие как "Каталог товаров", "О нас", "Контакты" и другие, в зависимости от функционала. Для наполнения каталога перейдите в раздел "Мои товары", где можно создать каталоги, добавить товары с указанием названия, описания, цены и загрузить изображения. Если приложение предназначено для интернет-магазина, настройте корзину и способы получения заказов, например, через Telegram или электронную почту.

Для расширения функциональности добавьте дополнительные модули, такие как регистрация пользователей, формы записи или система уведомлений. Когда структура и функционал будут готовы, воспользуйтесь функцией предварительного просмотра, чтобы проверить, как приложение выглядит на различных устройствах. В случае необходимости внесите изменения в дизайн или функционал.

После завершения работы нажмите "Собрать приложение". По итогам сборки вы получите APK-файл для Android или AAB-файл для публикации в Google Play. Для работы с платформой iOS нужно следовать инструкциям сервиса. Перед публикацией протестируйте приложение на собственных устройствах, подготовьте иконки и описания, которые потребуются для размещения в Google Play или App Store.

**3.2Организация данных**

Определив атрибуты для каждой сущности и установив связи между ними, получили следующую схему данных (рисунок 1).



Рисунок 1 – Схема-данных

В данной БД представлено 7 сущностей. Описание каждой из них представлено в таблице 5.

Таблица 5 – Сущности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Сущность | Имя поля | Тип данных | Описание |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Магазин | ID\_магазина | Числовой | Содержит информацию о магазине |
| Название | Текстовый |
| Адрес | Текстовый |
| Режим работы | Текстовый |
| 2 | Товар | ID товара | Числовой | Содержит информацию о товарах |
| ID категории | Числовой |
| ID акции | Числовой |
| ID магазина | Числовой |
| Название | Текстовый |
| Описание | Текстовый |
| Наличие в магазине | Текстовый |
| Цена | Текстовый |
| Цена со скидкой | Текстовый |
| Фото | Текстовый |
| 3 | Элемент брони | ID элемента брони | Числовой | Содержит информацию о бронировании |
| ID товара | Числовой |
| Количество единиц | Числовой |
| 4 | Категории | ID категории | Числовой | Содержит информацию о категориях |
| Название | Текстовый |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Таблица 5 продолжение | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 5 | Акции | ID акции | Числовой | Содержит информацию о акциях |
| Срок проведения | Текстовый |
| Размер скидки | Числовой |
| 6 | Элемент заказа | ID заказа | Числовой | Содержит информацию об элементе заказа |
| ID элемента заказа | Числовой |
| ID товара | Числовой |
| Стоимость товара | Числовой |
| Количество единиц | Числовой |
| 7 | Заказ | ID заказа | Числовой | Содержит информацию о заказе |
| ID пользователя | Числовой |
| Статус заказа | Текстовый |
| Адрес доставки | Текстовый |
| Примечания | Текстовый |
| 8 | Пользователь | ID пользователя | Числовой | Содержит информацию о пользователе |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
|  |  | Адрес | Текстовый |  |
| e-mail | Текстовый |
| 9 | Бронь | ID пользователя | Числовой | Содержит информацию о бронировании |
| ID брони | Числовой |
| Местонахождение | Текстовый |
| Время брони | Текстовый |

1. **Тестирование**

Каждый тест выполнялся пошагово: сначала проверялась работа отдельных элементов интерфейса, таких как кнопки, формы и ссылки. Затем тестировались основные сценарии использования, включая добавление товара в корзину, оформление заказа и авторизация пользователей.

Для проверки корректности работы различных функциональных модулей использовались тестовые учетные записи пользователей и предварительно созданные данные, такие как фейковые заказы и информация о товарах. Все тесты выполнялись вручную в реальных условиях работы сайта с использованием популярных браузеров, включая Google Chrome и Mozilla Firefox.

В процессе тестирования выявлялись ошибки, которые записывались в отдельный журнал для дальнейшего исправления. После устранения ошибок тесты повторялись, чтобы убедиться в их корректной работе и отсутствии регрессионных багов (приложение Д).

1. **Руководство пользователя**
   1. **Запуск программы**

Открываем приложение «IvantexShop».

* 1. **Инструкция по работе с программой**

Для того чтобы запустить мобильное приложение, оно должно быть установлено на Android устройство. Открывает приложение нажав на иконку «IvantexShop» (рисунок 2).



Рисунок 2 – Иконка приложения

После нажатия открывается главный экран (рисунок 3).

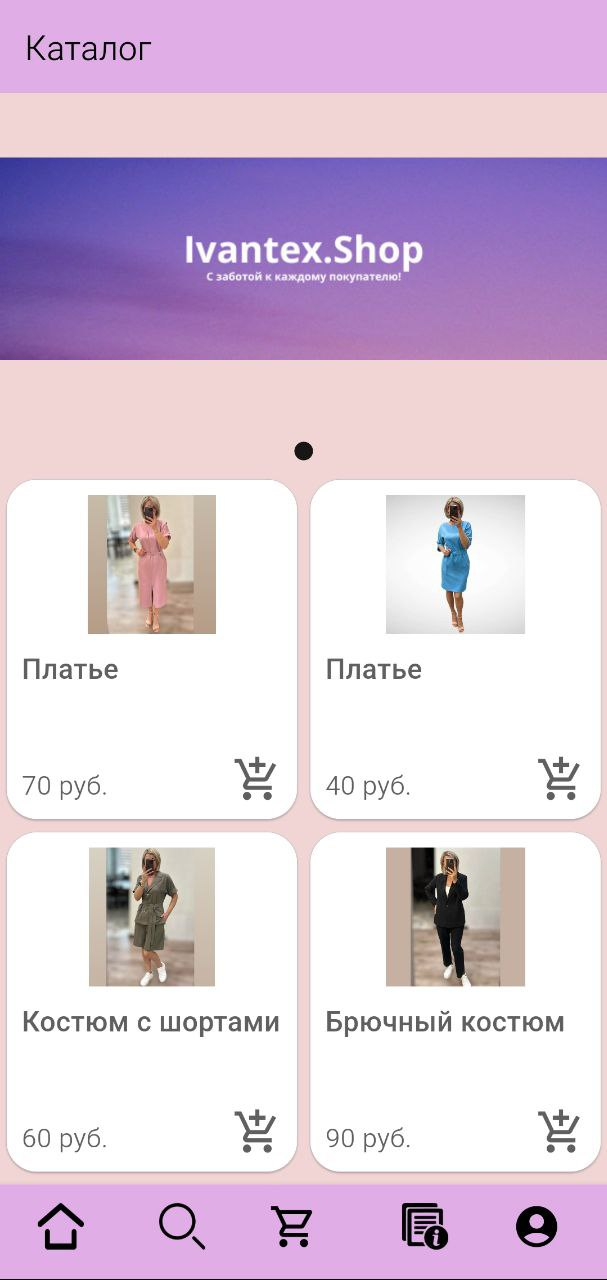


Рисунок 3 – Главный экран

После открытия мобильного приложения, вы можете просматривать каталог товаров, добавлять товары в корзину а так же удалять их из корзины (рисунок 4)

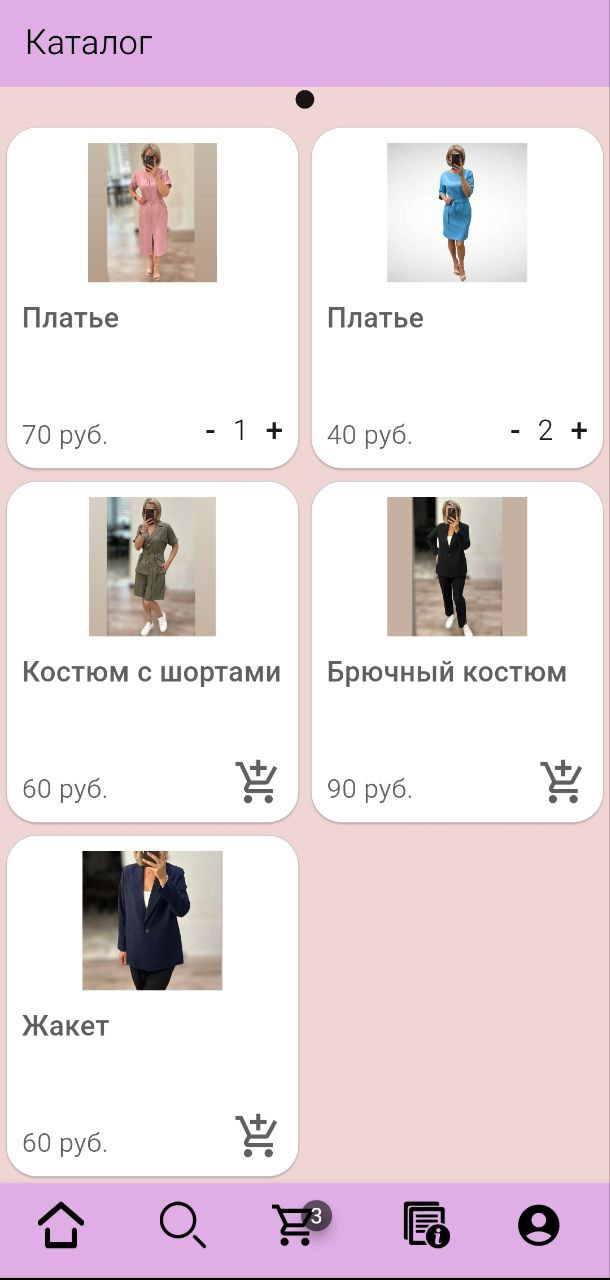


Рисунок 4 – Работа на главной странице

Также мы можем перейти на страницу поиска для выбора категории товаров (рисунок 5).

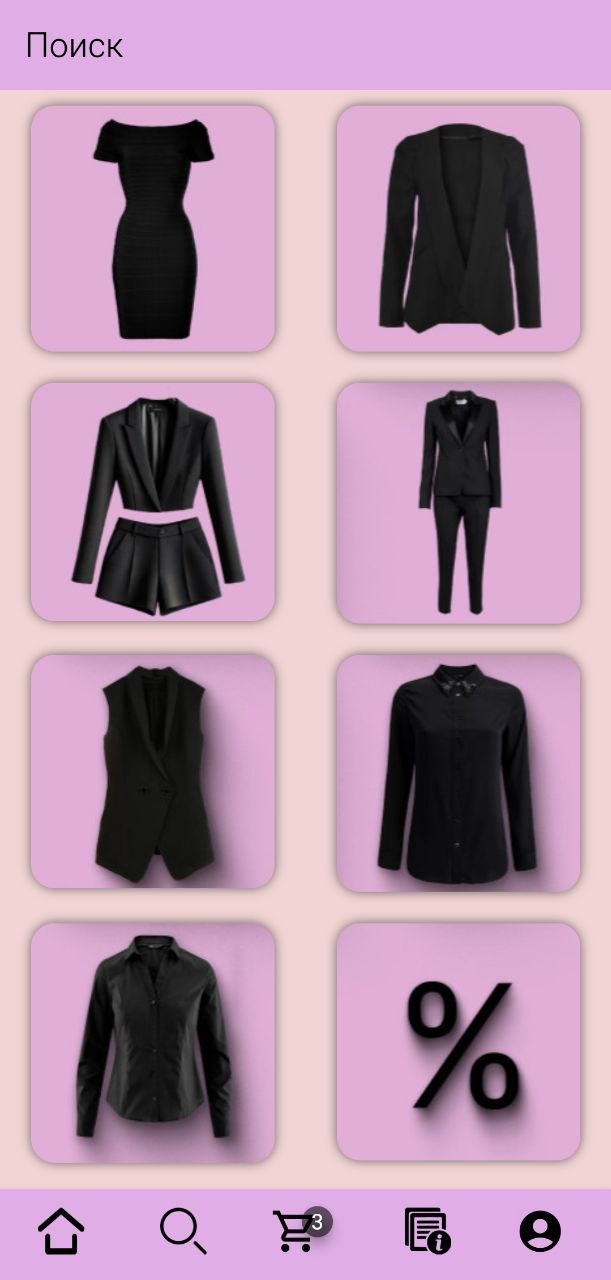


Рисунок 5 – Страница поиска

Рассмотрим категорию платья. Для этого нажимаем на иконку с платьем и попадаем на страницу категории «Платья» (рисунок 6)

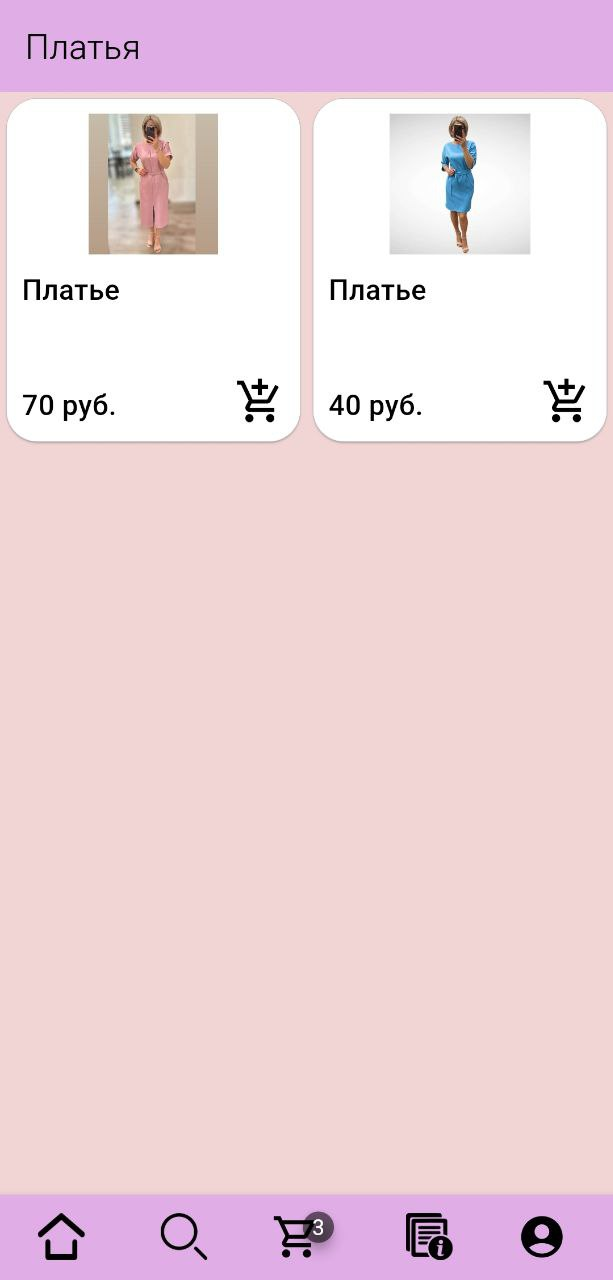


Рисунок 6 – Категория «Платья»

При нажатии на иконки тележки переходим в корзину с товарами где можно редактировать количество товаров и оформить заказ (рисунок 7).

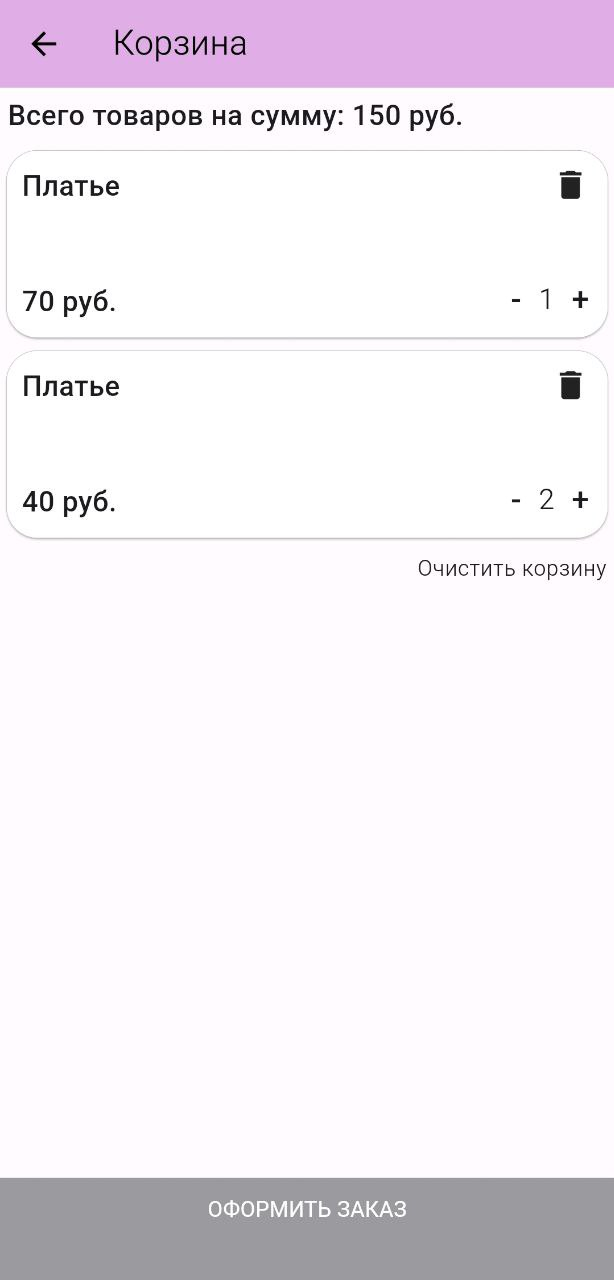


Рисунок 7 – Корзина

При нажатии на кнопку оформить заказ вылетает форма для заполнения личной информации пользователя для получения заказа (рисунок 8).

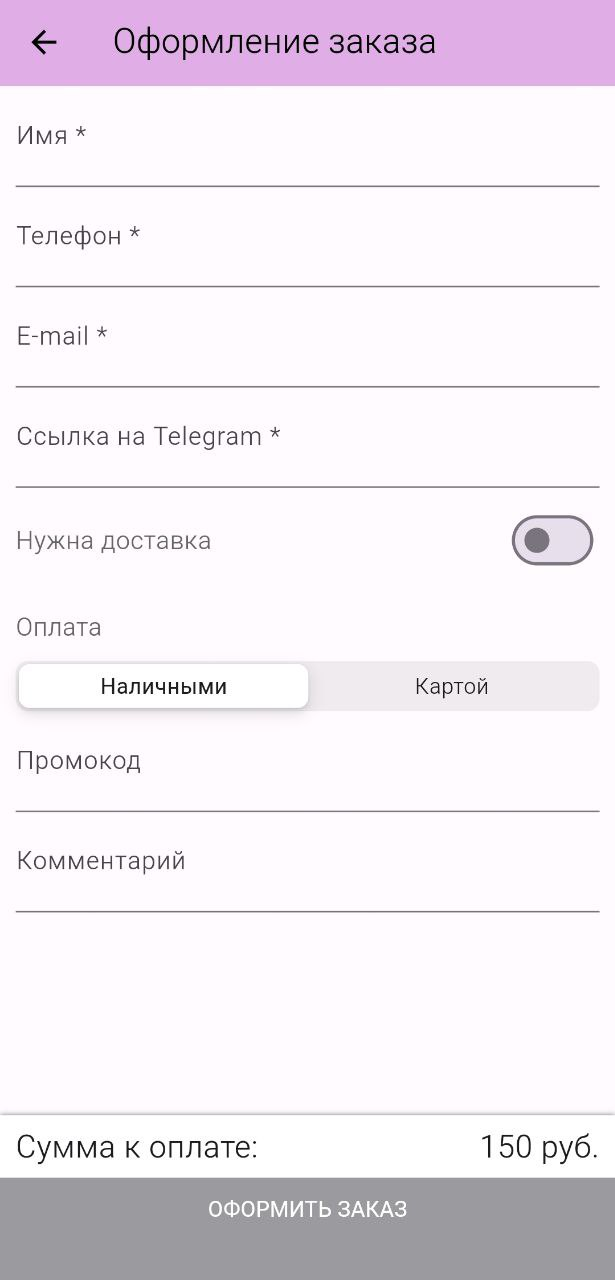


Рисунок 8 – Форма оформления заказа

Также есть информационная страница при нажатии на иконку листов попадаем на нее (рисунок 9)

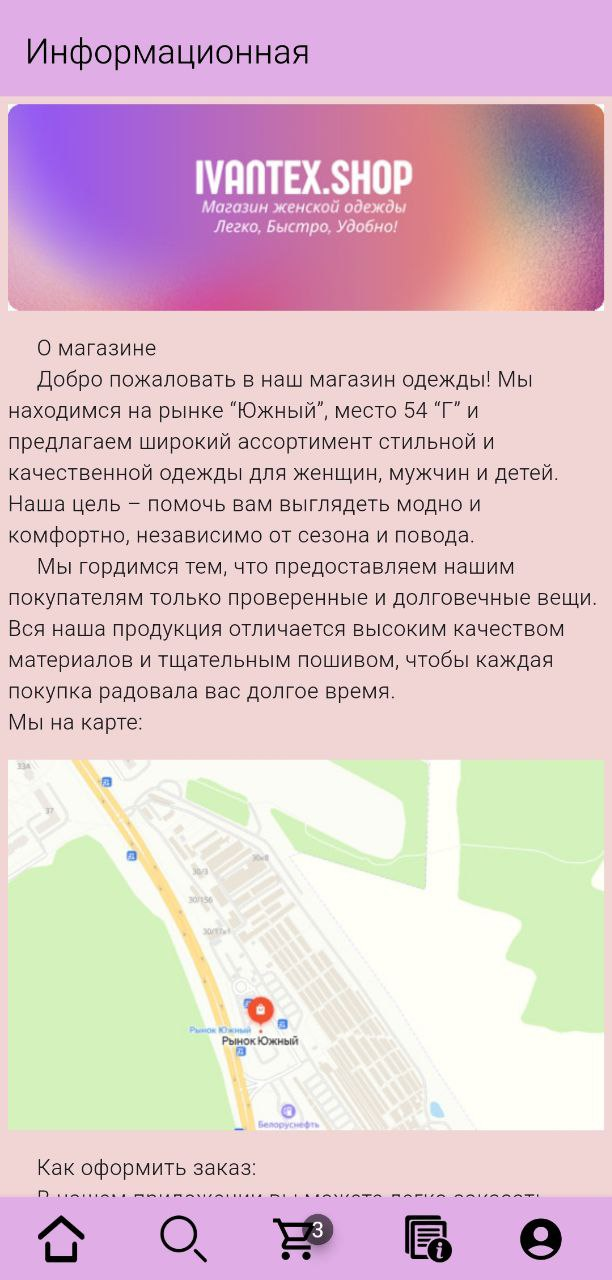


Рисунок 9 – Информационная страница

При нажатии на иконку с человечком попадаем в личный кабинет (рисунок 10)



Рисунок 10 – Личный кабинет

В личном кабинете можно редактировать информацию о пользователе нажав на кнопку «Настройки профиля» (рисунок 11).

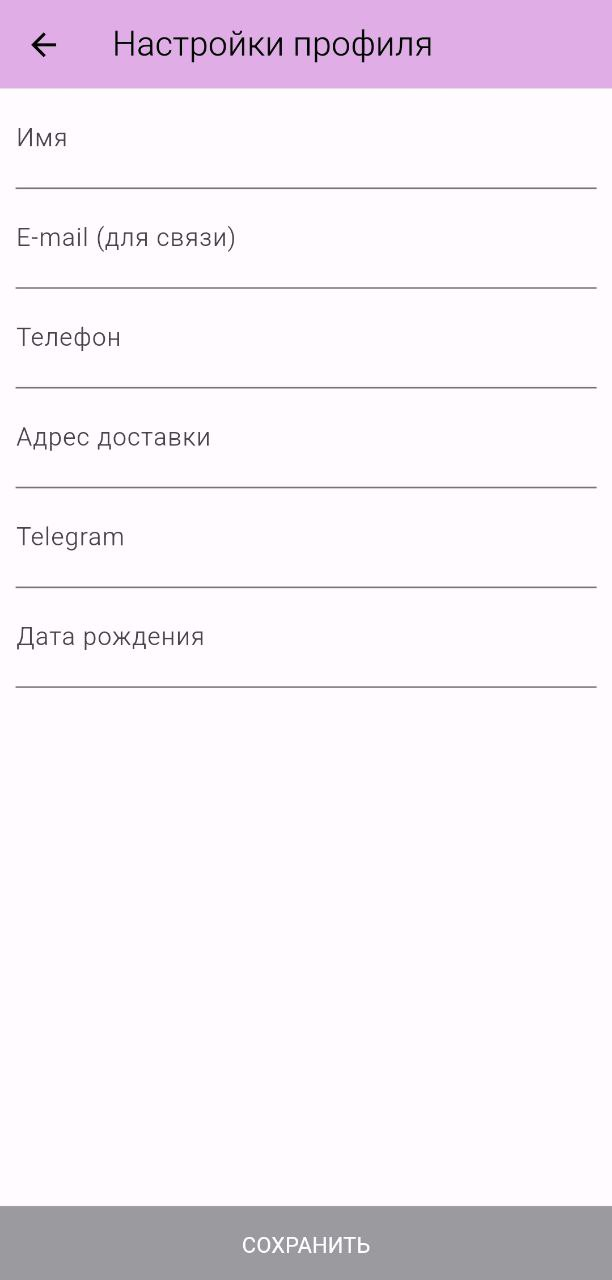


Рисунок 11 – Настройки профиля

Также из личного кабинета можно просмотреть историю заказа нажав на кнопку «История заказов»

* 1. **Завершение работы с программой**

Закрываем сайт «IvantexShop».

**Заключение**

Разработка приложения на тему: «Разработка интернет-магазина одежды IvantexShop», направленная на создание удобного и функционального мобильного приложения для продажи женской одежды, была выполнена с использованием языков программирования Java и Kotlin.

Для разработки приложения использовались следующие технологии и инструменты:

* Java, Kotlin – для реализации логики и функциональности приложения.
* XML – для описания пользовательского интерфейса.
* Android Studio – для разработки и тестирования приложения.
* Firebase – для работы с базой данных и авторизацией пользователей.
* Git – для контроля версий и управления разработкой.
* Microsoft Word 2021 – для написания документации по проекту.

Использование данных технологий и инструментов позволило создать полноценное мобильное приложение, включающее основные функции, такие как:

* Регистрация и авторизация пользователей.
* Добавление и редактирование товаров.
* Просмотр и фильтрация каталога товаров.
* Управление заказами и статусами покупок.
* Создание адаптивного интерфейса, удобного для работы на мобильных устройствах.

Степень соответствия проектных решений заданию: приложение было выполнено в соответствии с поставленными требованиями.

Проект демонстрирует возможности современных технологий для создания качественного и функционального продукта, который может быть дополнен и расширен в будущем для улучшения пользовательского опыта и добавления новых функций.

**Список использованных источников**

1. ChatGPT