МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ И МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ ТОВАРАМИ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Байрамова Шамистана Мохуббат оглы

Научный руководитель
к. ф.-м. н., доцент _____ А. С. Иванова

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2025

ВВЕДЕНИЕ

В современном мире цифровизация бизнеса играет важную роль в обеспечении конкурентоспособности и повышения эффективности взаимодействия с клиентами. Веб-приложения становятся основным инструментом для улучшения клиентского опыта, предоставляя пользователям удобные и доступные сервисы для решения разнообразных задач. Особенно важными являются приложения для электронной коммерции, которые обеспечивают продажи товаров, удобное управление заказами и поддержку взаимодействия между покупателями и продавцами.

Целью данной работы является разработка веб-приложения и мобильного приложения под Android для управления товарами и поиска товаров с возможностью авторизации пользователей. Веб-приложение будет интегрировать различные элементы, которые обеспечат не только удобство работы с товарами, но и высокий уровень безопасности данных пользователей.

Система будет включать функционал для добавления, редактирования и удаления и поиска товаров. Особое внимание уделяется безопасности данных, что включает использование аутентификации пользователей и защищённых каналов передачи данных.

Для достижения поставленной цели необходимо решить следующие задачи:

- Разработать способы безопасного хранения и обработки данных пользователей с использованием реляционных баз данных, таких как PostgreSQL.
- Исследовать методы авторизации и аутентификации пользователей с использованием Spring Security.
- Разработать веб-приложение для управления товарами с возможностью поиска по геолокации.
- Разработать Rest API для подключения к серверу сторонних сервисов.
- Разработать мобильное приложение для управления товарами на Android и подключится к серверу с пмощью API.

Разработанные приложения будут обеспечивать простой и интуитивно понятный интерфейс для управления товарами, при этом учитывая все аспекты безопасности и эффективного управления данными. Этот проект позволит продемонстрировать возможности технологий Kotlin, Android Studio, Spring Boot, PostgreSQL и других современных инструментов для создания масштабируемых

и безопасных веб-приложений.

Структура и объём работы. Для решения поставленных задач выполнена выпускная квалификационная работа, включающая в себя введение, 3 основные главы, заключение, список использованных источников из 20 наименований и 5 приложений. Работа изложена на 87 страницах.

Первый раздел «Обзор используемых технологий для разработки веб-приложения» посвящен описанию основных определений используемых в работе, а также теоретической информации, касающейся инструментов веб-разработки.

Для разработки веб-приложения по управлению товарами используется ряд современных технологий и инструментов, обеспечивающих высокую производительность, безопасность и удобство использования. Основные технологии включают в себя Spring MVC, Spring Boot, PostgreSQL, FreeMarker, Spring Security, Spring Data JPA и DAO.

Spring Framework — это мощный и гибкий фреймворк для разработки enterprise-приложений на Java. Он предлагает комплексную инфраструктуру, позволяющую создавать масштабируемые, модульные и легко тестируемые приложения, основанные на ключевых принципах инверсии управления (IoC) и внедрения зависимостей (DI).

Для внедрения зависимостей используется аннотация @Autowired, которая дает Spring контейнеру понять метод и место внедрения зависимости. Создается объект, который в своем конструкторе ссылается на класс реализующий инверсию управления.

Spring Boot — это модуль экосистемы Spring, упрощающий создание Java-приложений за счёт автоконфигурации, starter-пакетов и встроенного сервера. В отличие от классического Spring, он минимизирует ручную настройку, автоматически подбирая зависимости и параметры на основе окружения.

Starter-пакеты (например, spring-boot-starter-web, spring-boot-starter-data-jpa) включают всё необходимое для конкретных задач — веб-разработки, работы с БД, безопасности, избавляя от поиска совместимых библиотек. Spring Boot также предоставляет встроенные серверы (Tomcat, Netty), метрики, health-checks и конфигурацию через application.properties, делая развёртывание и разработку максимально простыми.

FreeMarker — это Java-шаблонизатор для генерации HTML, XML, email и

других текстовых форматов, используемый в Spring MVC для рендеринга представлений. Он работает как мост между моделью и представлением, позволяя не только выводить данные, но и передавать их обратно в модель.

Spring Security — это мощный модуль экосистемы Spring, предназначенный для реализации механизмов аутентификации и авторизации в Java-приложениях. Он предоставляет гибкие инструменты для защиты веб-приложений, REST API и микросервисов, позволяя контролировать доступ к ресурсам на основе ролей и прав пользователей.

Фреймворк поддерживает различные методы аутентификации: от простой HTTP Basic и Form-based до современных решений, таких как OAuth2, JWT-токены, LDAP и SAML. Spring Security легко интегрируется с другими технологиями, включая Spring Boot, и предлагает готовые решения для CSRF-защиты, CORS-политик и безопасности методов.

Spring MVC — это компонент Spring Framework, реализующий классическую архитектуру MVC для веб-разработки. Он разделяет приложение на три основных слоя: модель данных, представление и контроллеры, что делает код чище и проще в поддержке.

Главное преимущество Spring MVC — глубокая интеграция с экосистемой Spring, включая безопасность, доступ к данным и REST-сервисы. Это позволяет создавать сложные веб-приложения с минимальной ручной настройкой, сосредоточившись на основной функциональности.

Spring Data JPA — это мощный инструмент экосистемы Spring, который кардинально упрощает работу с базами данных через JPA. Он избавляет разработчиков от написания рутинного кода, автоматизируя стандартные операции с данными и предоставляя удобные абстракции для взаимодействия с различными СУБД.

Модуль идеально интегрируется с другими компонентами Spring, такими как Spring Boot и Spring MVC, обеспечивая комплексное решение для работы с данными в enterprise-приложениях.

Retrofit — это современная библиотека от компании Square, созданная специально для работы с HTTP-запросами в Android и Java-приложениях. Она превращает REST API в удобные Java-интерфейсы, автоматизируя процесс сетевого взаимодействия и избавляя разработчиков от рутинной работы с низкоуровневыми HTTP-запросами.

Библиотека особенно популярна в Android-разработке, где её лаконичный синтаксис (аннотации вроде @GET, @POST) и поддержка RxJava/Coroutines значительно упрощают асинхронные вызовы к серверу. Retrofit легко интегрируется с другими компонентами экосистемы Square (OkHttp) и становится стандартом для сетевых операций в мобильных приложениях.

Gson — это популярная библиотека от Google для преобразования JSON-данных в Java/Kotlin-объекты и обратно. Она стала стандартом де-факто в Android-разработке благодаря простоте использования и высокой производительности.

Библиотека автоматически сериализует объекты в JSON-строки и парсит JSON обратно в объекты, избавляя разработчиков от ручного разбора данных. Gson поддерживает сложные сценарии: работу с дженериками, кастомные адаптеры для нестандартных типов данных, обработку null-значений и гибкие настройки форматов дат.

Второй раздел «Описание разработки веб-приложения» посвящен детальному описанию кодовой базы инструментов веб-разработки.

При создании торговой платформы Tovarisk основной акцент был сделан на построении надежной серверной части с использованием современного стека технологий Java. В качестве основного фреймворка был выбран Spring Boot, что позволило создать масштабируемое приложение с поддержкой всех необходимых функций электронной торговли.

Архитектура приложения построена по принципу многоуровневого разделения ответственности, где каждый компонент выполняет строго определенные задачи. Взаимодействие с базой данных реализовано через слой DAO, что обеспечивает гибкость при работе с данными и позволяет легко модифицировать логику доступа к хранилищу без изменения бизнес-логики. Особенностью разработанной платформы является система управления изображениями с автоматической оптимизацией размера файлов. Реализован механизм предварительного просмотра и хранения нескольких изображений для каждого товара, что улучшает пользовательский опыт при работе с платформой. Система безопасности, построенная на Spring Security, обеспечивает надежную защиту пользовательских данных и управление доступом к различным частям приложения.

В приложении реализована гибкая система поиска и фильтрации товаров, позволяющая пользователям быстро находить нужные предложения по различ-

ным критериям, включая название, описание и местоположение. Интеграция с фронтенд-частью осуществляется через Spring MVC.

Отдельное внимание было уделено оптимизации производительности. Реализована система кэширования часто запрашиваемых данных, а также механизмы эффективной работы с базой данных через подготовленные запросы. Это позволяет системе эффективно работать даже при высоких нагрузках и большом количестве одновременных пользователей. Разработка велась с учетом возможности дальнейшего масштабирования и добавления новых функций. Модульная структура приложения и использование современных практик программирования позволяют легко расширять функциональность системы без необходимости существенного изменения существующего кода.

Для обеспечения качества кода и надежности работы приложения использовались инструменты автоматической сборки и управления зависимостями Maven. Это позволило организовать эффективный процесс разработки и поддержки приложения.

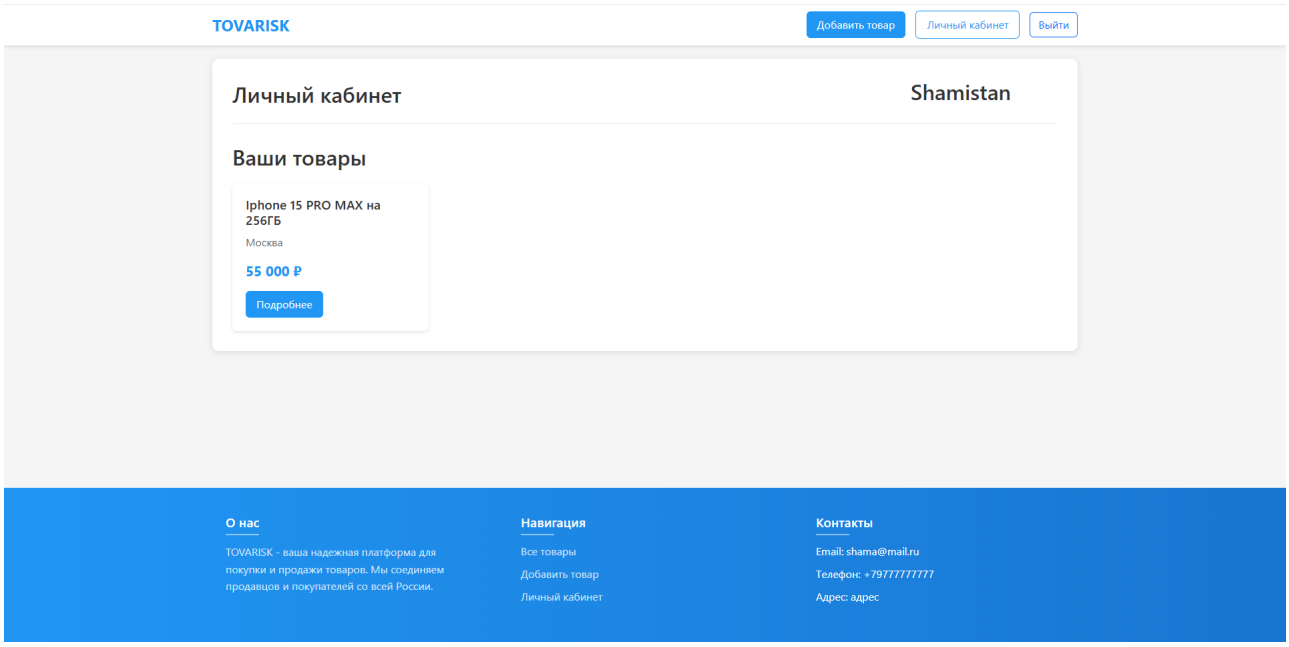

Рисунок 1 – Страница пользователя

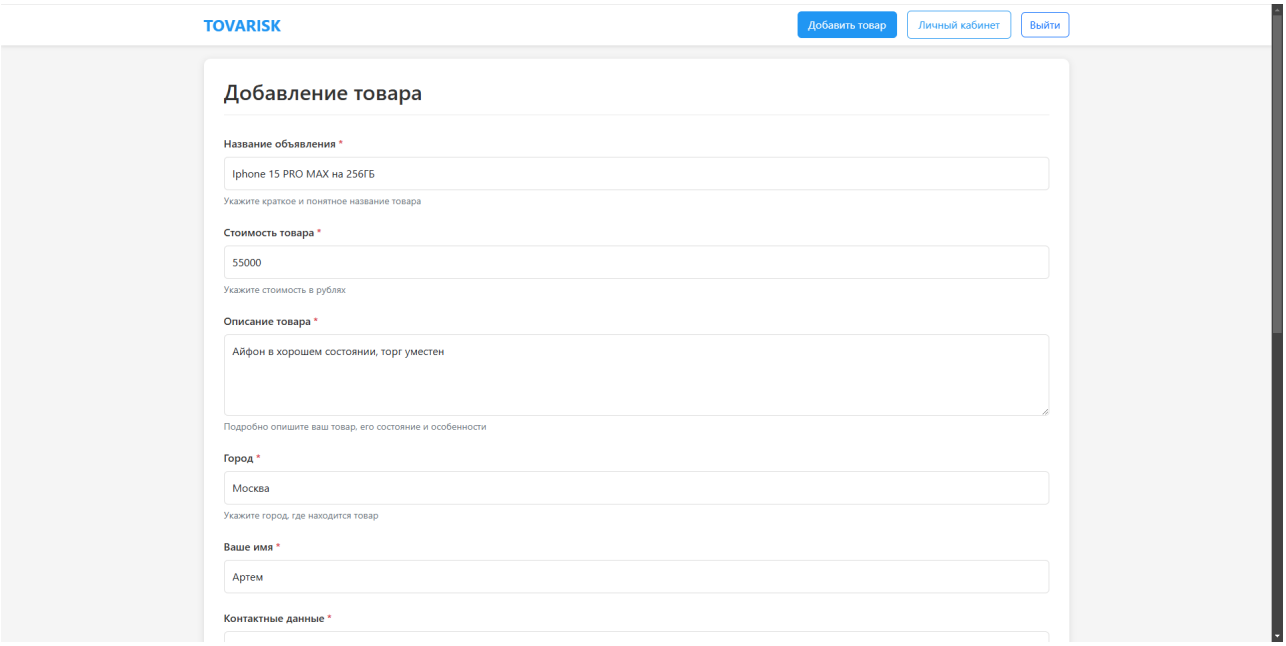

Рисунок 2 – Страница для добавления товара

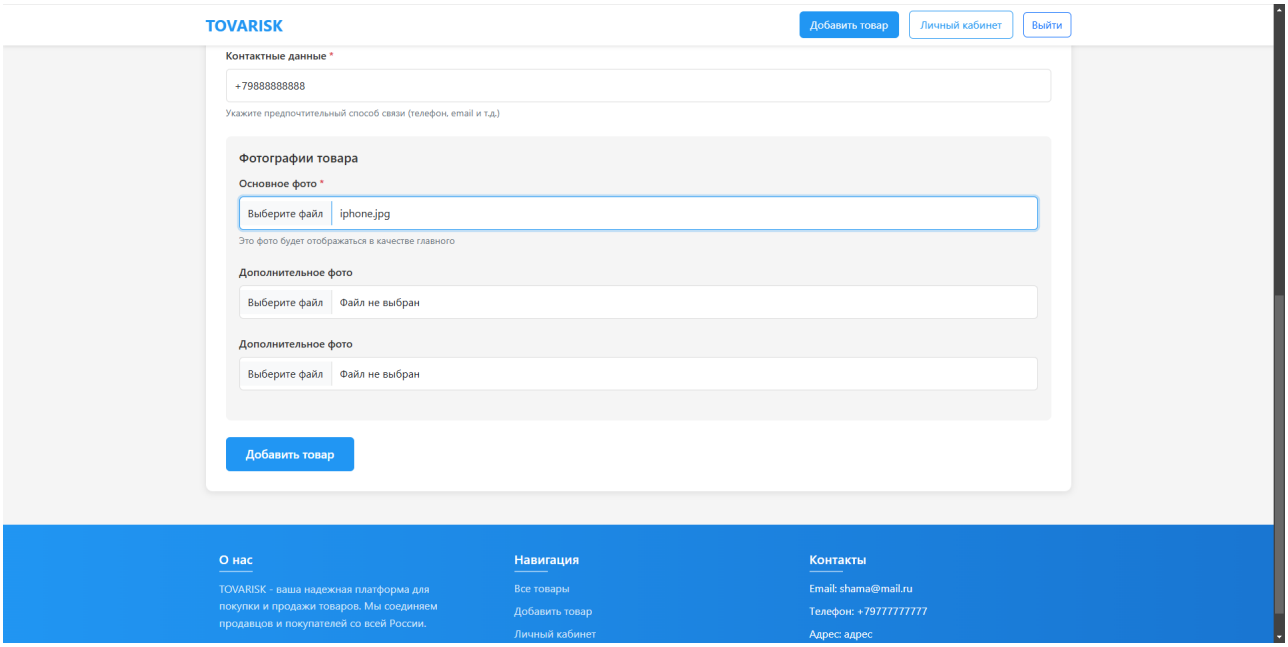

Рисунок 3 – Страница для добавления товара

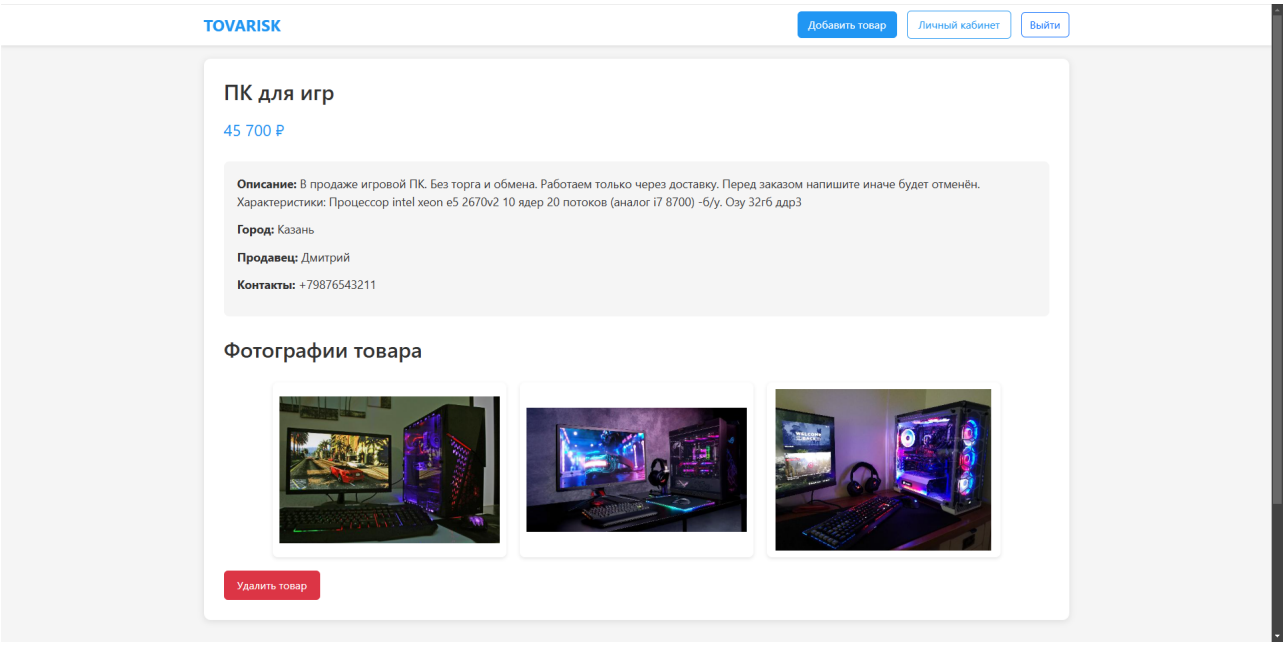

Рисунок 4 – Страница для просмотра определенного товара

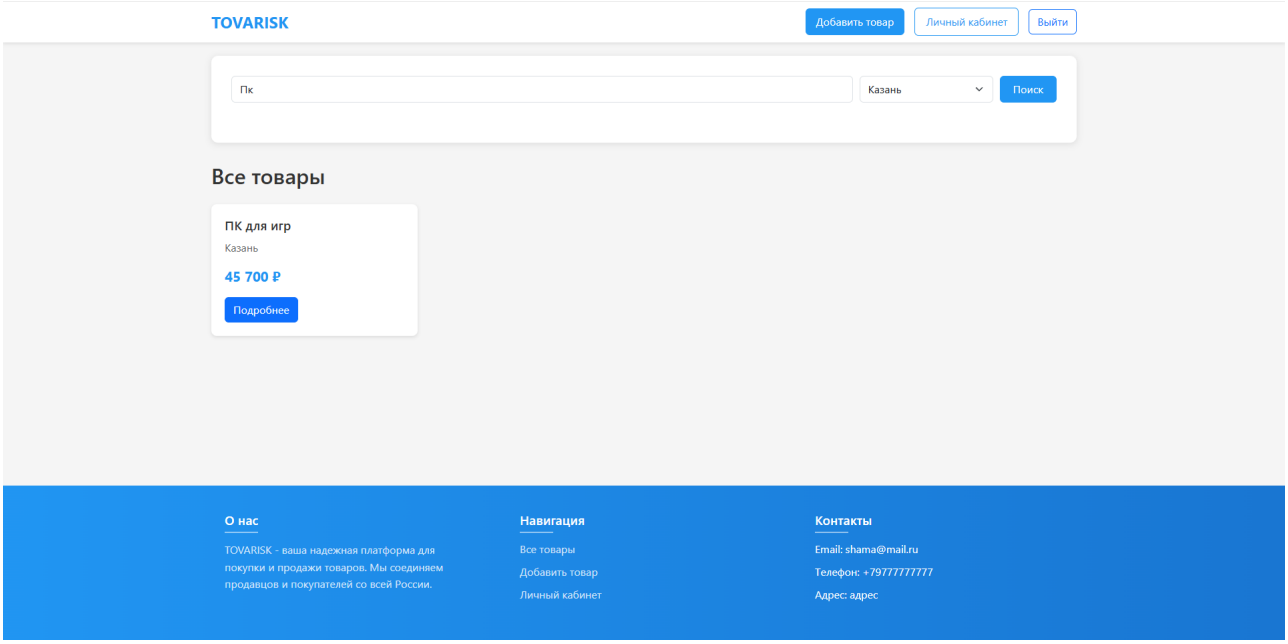

Рисунок 5 – Результат поиска конкретного товара

Третий раздел «Описание разработки мобильного приложения» посвящен детальному описанию кодовой базы инструментов мобильной разработки. При разработке мобильного приложения Tovarisk для платформы Android основной фокус был направлен на создание интуитивно понятного и отзывчивого пользовательского интерфейса с использованием современного стека технологий Kotlin. Приложение разработано с учетом последних рекомендаций

Material Design, что дает привлекательный внешний вид и удобство использования для пользователей.

Архитектура мобильного приложения построена с использованием принципов чистой архитектуры, где каждый компонент имеет четко определенную зону ответственности. Взаимодействие с серверной частью реализовано через Retrofit. Структура приложения разделена на логические модули: сетевой слой, модели данных, адаптеры для отображения списков и утилиты.

Реализован удобный механизм просмотра фотографий через ImagePagerAdapter, который обеспечивает плавную навигацию между изображениями товара. Система кэширования изображений и их оптимизация позволяют эффективно использовать ресурсы устройства и экономить мобильный трафик пользователей.

В приложении реализована комплексная система аутентификации пользователей с поддержкой регистрации и авторизации. Профильный раздел позволяет пользователям управлять своими данными и отслеживать активность на платформе. Также добавление товаров включает возможность загрузки нескольких фотографий и указания детальной информации о предложении.

Пользовательский интерфейс построен с использованием компонентов Android SDK и библиотек поддержки. RecyclerView с кастомными адаптерами реализует эффективное отображение списков товаров, a GridSpacingItemDecoration добавляет профессиональный внешний вид при отображении сетки элементов.

Для обеспечения качества кода и управления зависимостями использовался Gradle.

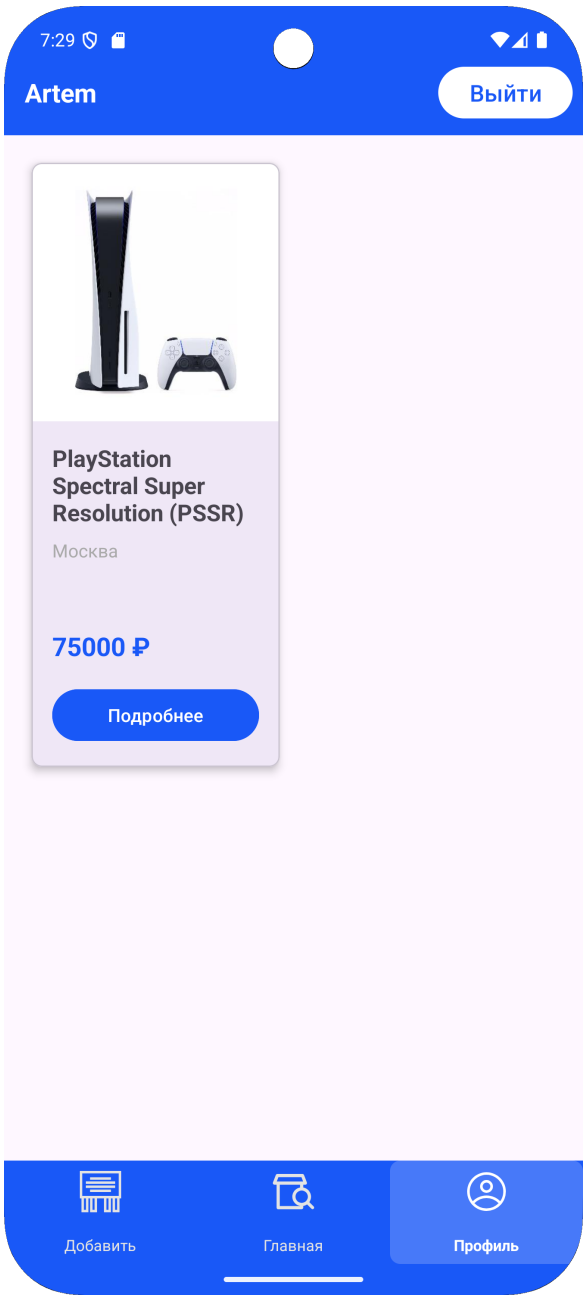

Рисунок 6 – Вид личного кабинета пользователя

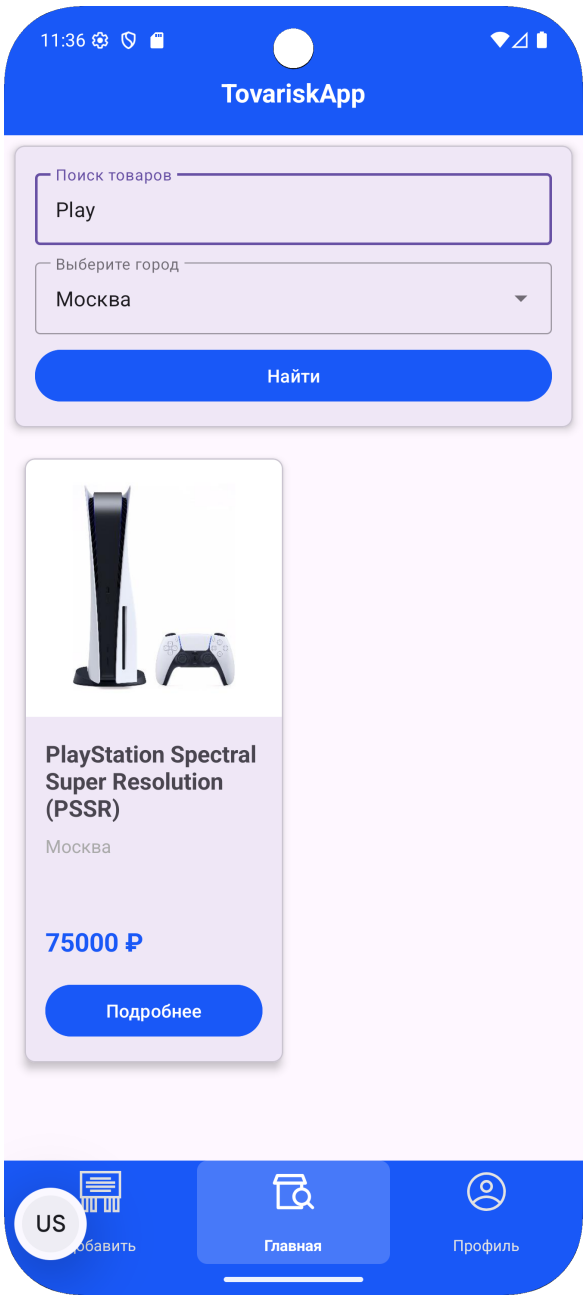

Рисунок 7 – Работа поиска товаров в приложении

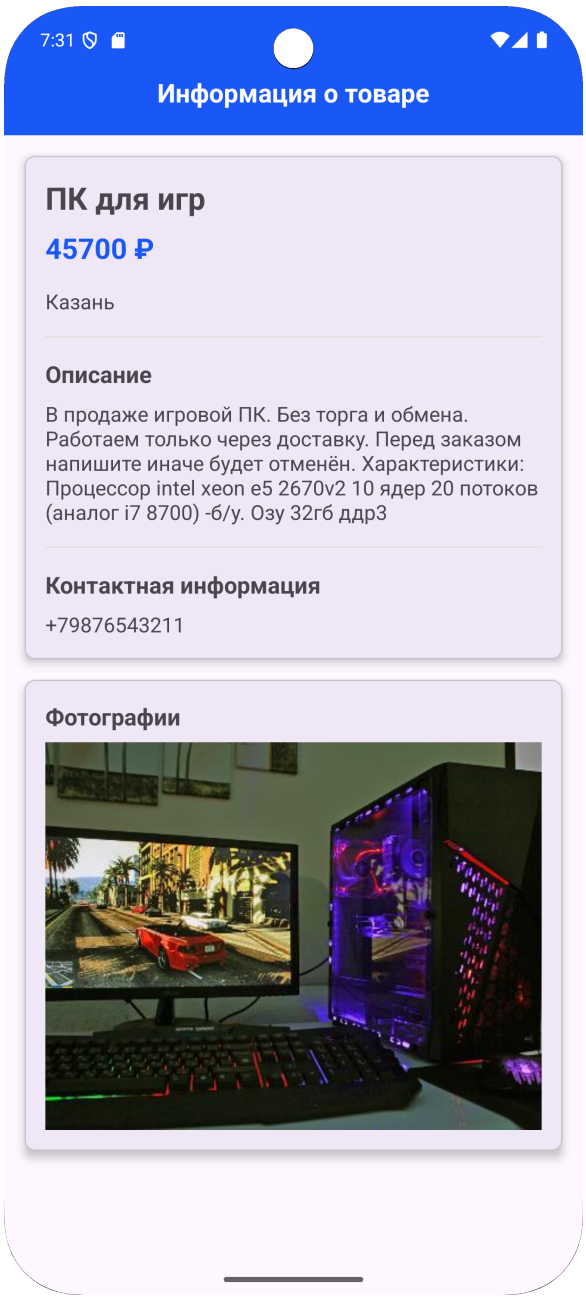

Рисунок 8 – Вид карточки товара

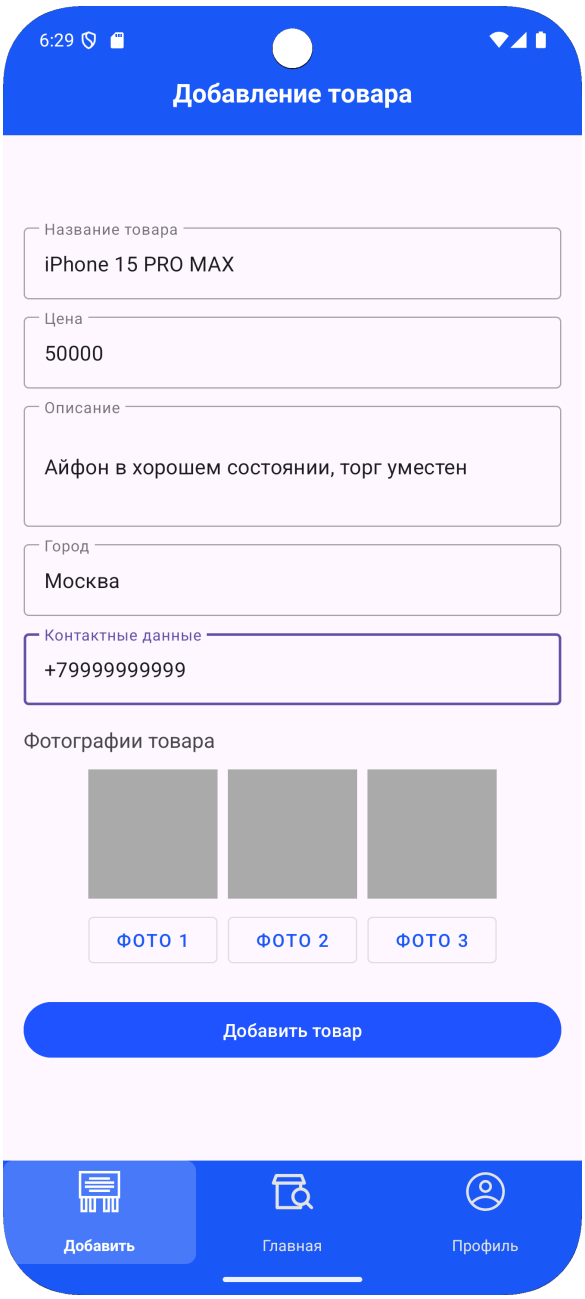

Рисунок 9 – Вид добавления товара

ЗАКЛЮЧЕНИЕ

В рамках выполненной работы был реализован комплексный подход к созданию распределенной информационной системы, состоящей из двух компонентов: мобильного приложения "Tovarisk"для платформы Android и веб-приложения, разработанного с использованием фреймворка Spring. Такой двусторонний подход позволил создать полноценную экосистему, обеспечивающую широкие возможности доступа к функционалу системы как через мобильные устройства, так и через веб-интерфейс.

Разработка серверной части на базе Spring Framework демонстрирует применение современных принципов построения корпоративных приложений. Использование Spring обеспечило надежную основу для создания масштабируемого и безопасного backend-решения, реализующего все необходимые бизнес-процессы. Архитектура веб-приложения, построенная на принципах REST, обеспечивает эффективное взаимодействие между клиентской и серверной частями системы, а также предоставляет унифицированный интерфейс для доступа к данным как со стороны веб-клиента, так и со стороны мобильного приложения.

Мобильное приложение, разработанное на Kotlin с использованием современных инструментов Android-разработки, органично дополняет веб-компонент системы. Модульная архитектура мобильного приложения обеспечивает эффективное взаимодействие с серверной частью и предоставляет пользователям удобный доступ к функционалу системы через мобильные устройства. Использование Kotlin как основного языка разработки мобильного приложения позволило создать надежное и типобезопасное решение, полностью соответствующее современным стандартам разработки для платформы Android.