Требования к проекту

1 Введение

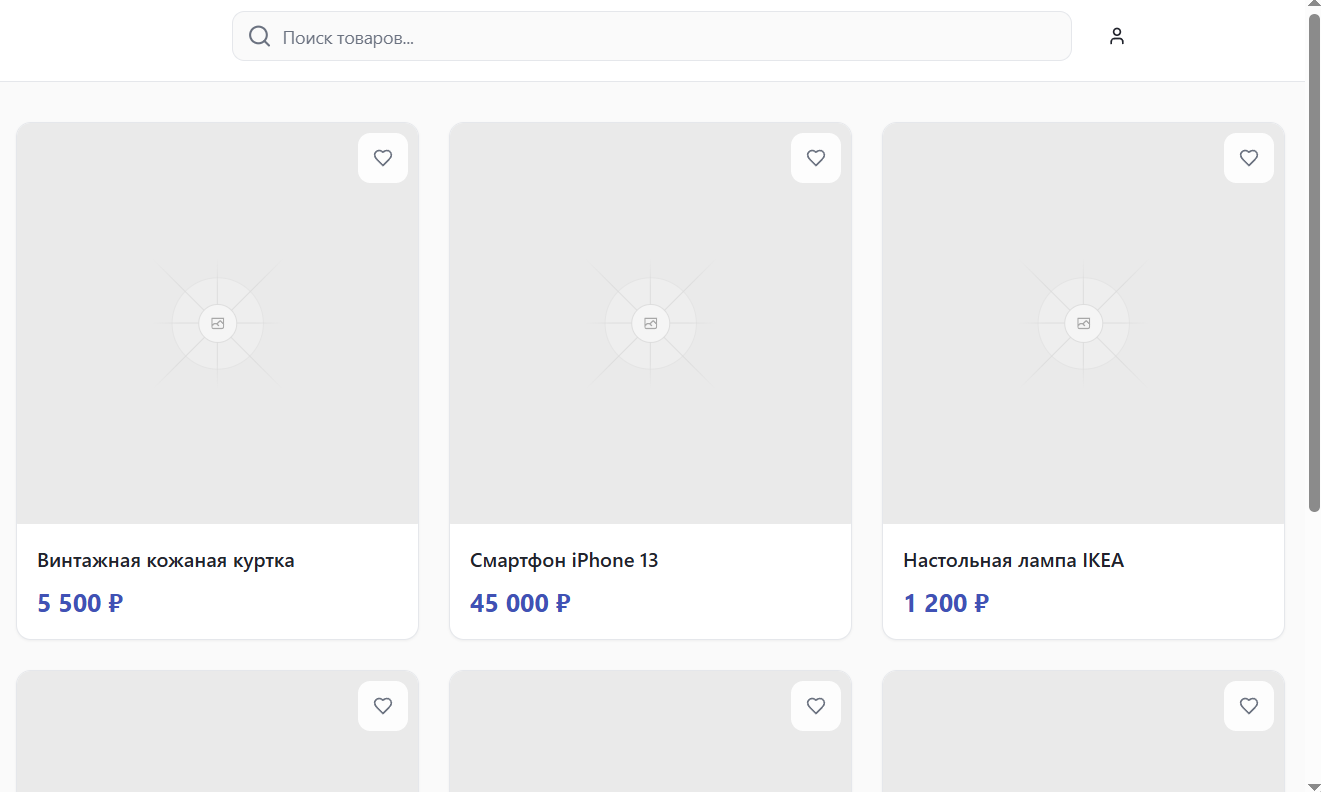
Контекст данного проекта заключается в создании современной и удобной онлайн-платформы для торговли между частными лицами, которая отвечает потребностям пользователей в простом и безопасном способе продавать и покупать самые разные товары. Продукт получает название "MarketPlace". В общих чертах, продукт будет представлять собой веб-приложение, где пользователи после регистрации и входа в систему могут выставлять свои товары на продажу, размещая фотографии, описания и контактные данные, а также просматривать и приобретать товары, выставленные другими пользователями. Ключевой функционал будет включать личный кабинет, управление объявлениями, поиск по каталогу и возможность добавлять товары в список избранного. Что касается границ проекта, продукт будет выполнять функции витрины и средства коммуникации, но не будет включать в себя встроенную систему проведения платежей, организацию доставки товаров или внутренний мессенджер для переписки, оставляя финальные этапы сделки на усмотрение пользователей, которые связываются друг с другом через указанные телефон и электронную почту.

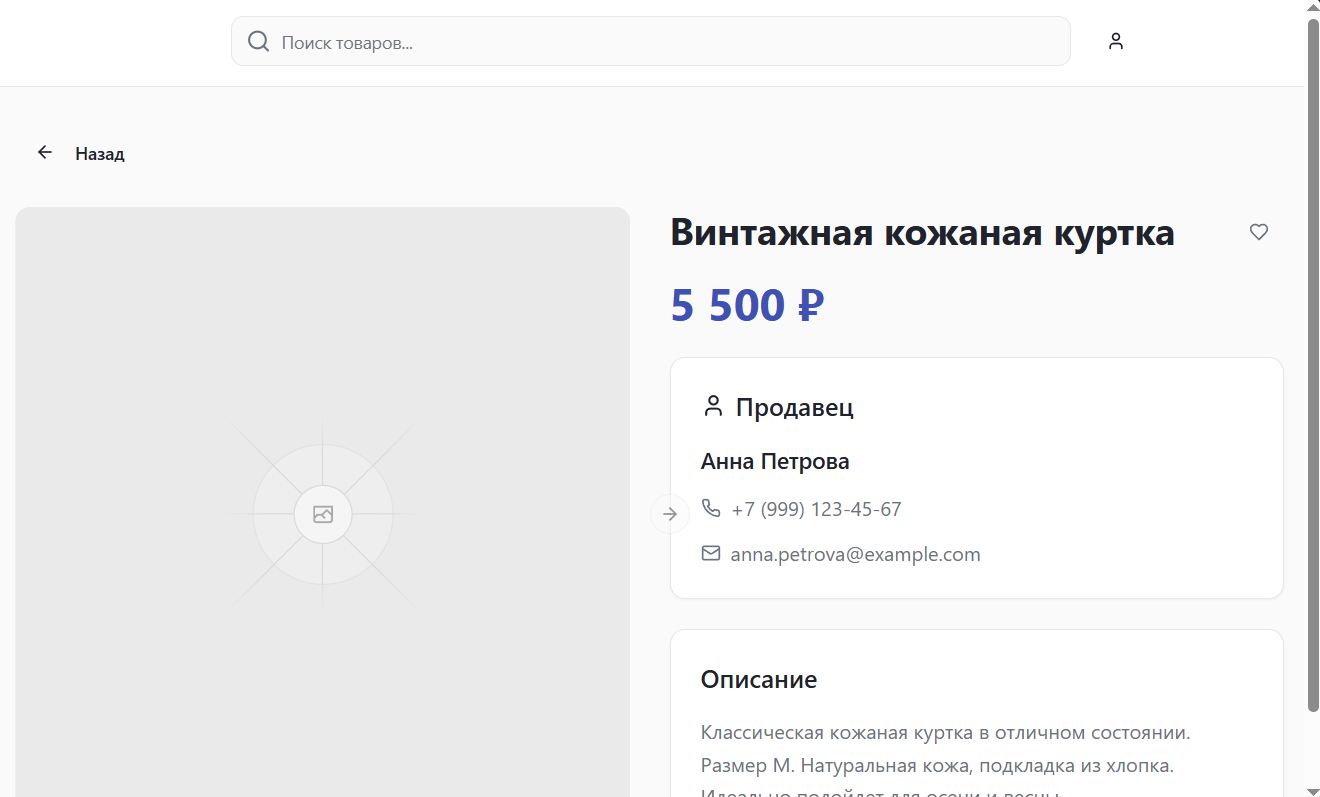
2 Требования пользователя

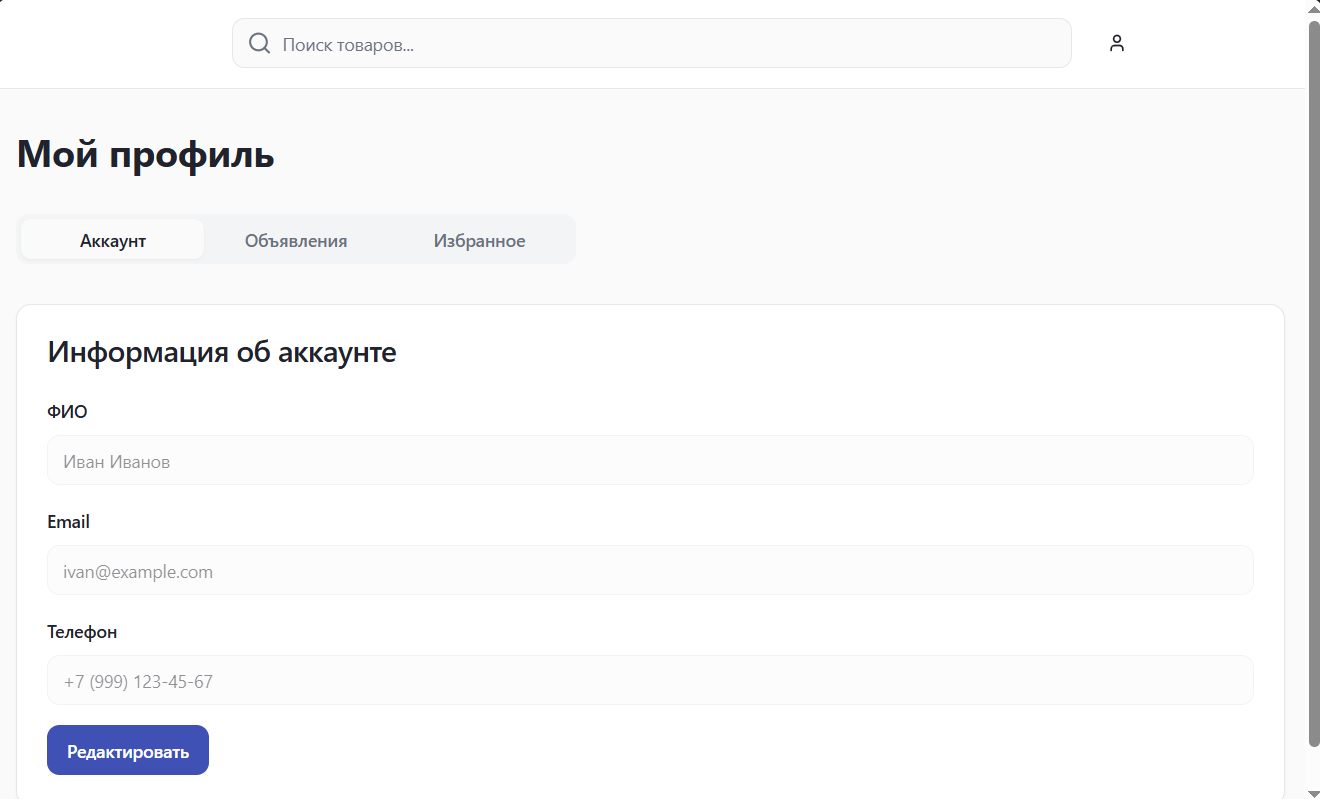
2.1 Программные интерфейсы

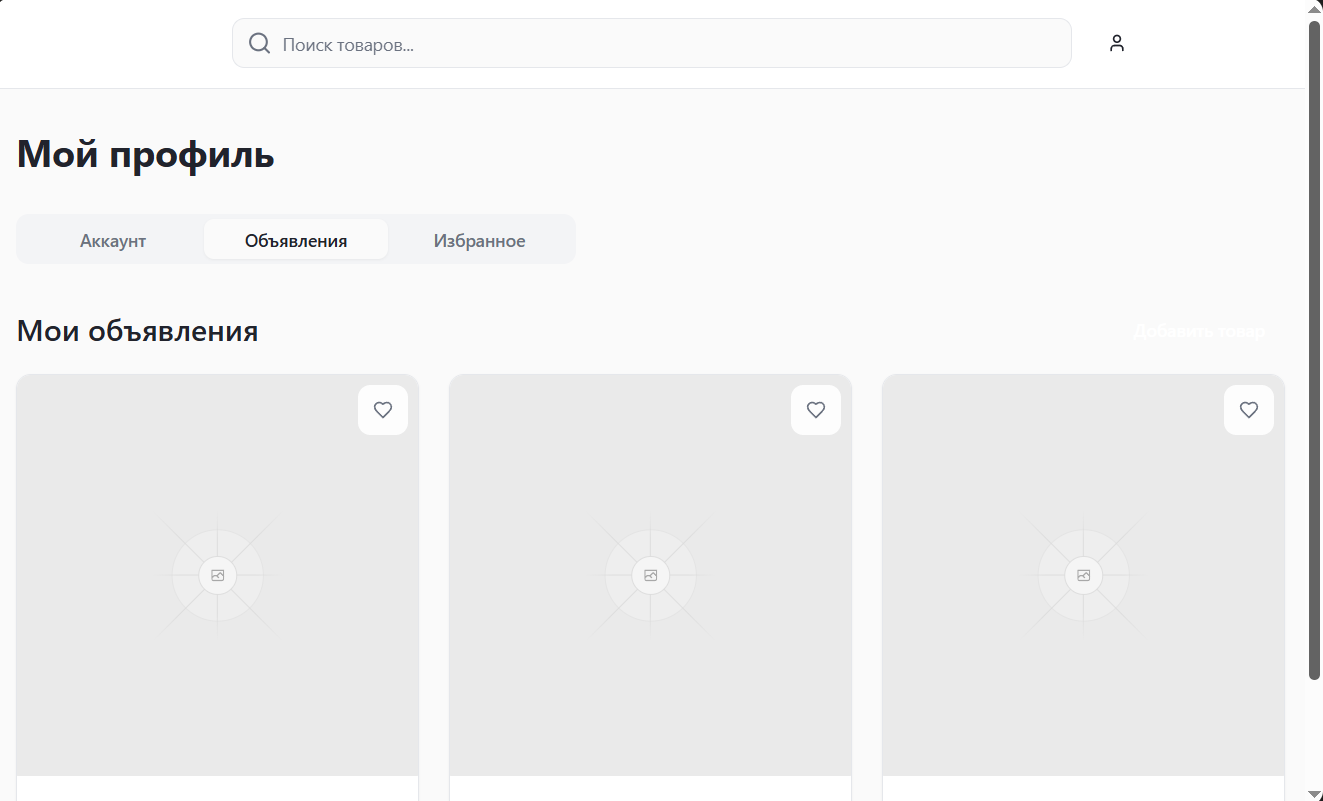
Продукт будет взаимодействовать с рядом внешних систем и библиотек для обеспечения своей полноценной функциональности. Бэкенд будет построен на основе фреймворка Spring Boot, который включает в себя Spring MVC для обработки веб-запросов, Spring Security для управления аутентификацией и авторизацией, и Spring Data JPA для упрощения работы с данными. Для хранения всей информации, такой как данные пользователей, объявления и списки избранного, будет использоваться реляционная система управления базами данных, например, PostgreSQL или MySQL. Взаимодействие с базой данных будет осуществляться через Hibernate в качестве реализации JPA. Для рендеринга пользовательского интерфейса на сервере будет применяться шаблонизатор Thymeleaf. Клиентская сторона будет использовать фреймворк Bootstrap для обеспечения адаптивного и визуально согласованного дизайна, а также JavaScript для реализации интерактивных элементов, таких как добавление в избранное без перезагрузки страницы. Для хранения загружаемых изображений товаров будет задействовано либо локальное файловое хранилище сервера, либо облачный сервис, такой как AWS S3.

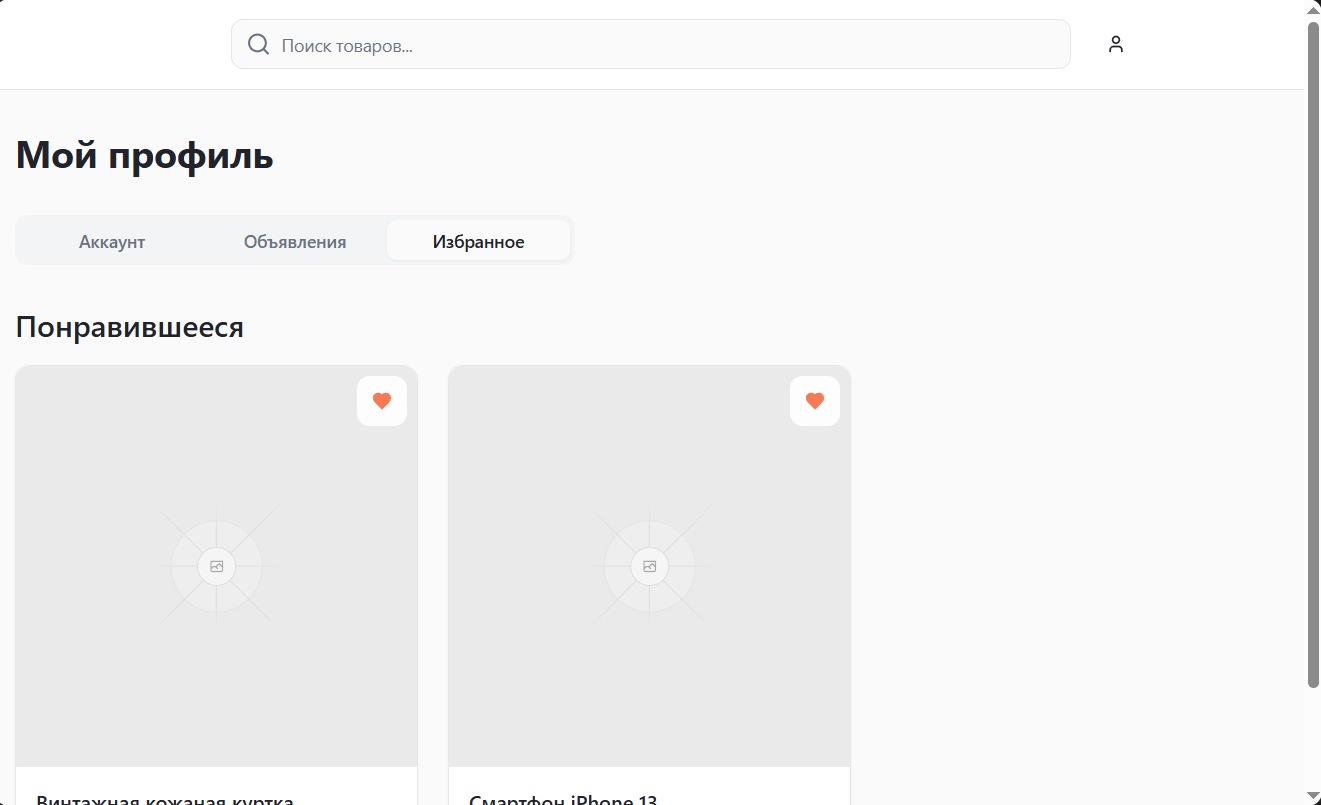
2.2 Интерфейс пользователя











2.3 Характеристики пользователей

Пользователей системы можно идентифицировать в две основные группы. Первую группу составляют продавцы — это физические лица, которые хотят продать ненужные или новые вещи. Их уровень технической грамотности, как правило, средний; они уверенно пользуются браузерами, умеют фотографировать товары на смартфон и заполнять онлайн-формы, имея опыт работы с социальными сетями и другими похожими платформами. Вторую группу представляют покупатели — это физические лица, которые ищут и приобретают товары. Их технический опыт может варьироваться от начального до продвинутого, но минимальным требованием является умение пользоваться поиском в интернете и базовыми функциями веб-сайтов.

2.4 Предположения и зависимости

На реализацию описанных требований могут повлиять несколько факторов. Ключевым предположением является наличие у конечных пользователей стабильного доступа в интернет и современного веб-браузера. Также предполагается, что пользователи будут действовать добросовестно, не пытаясь взломать систему или размещать незаконный контент. Техническая реализация зависит от стабильности и совместимости выбранных технологий, таких как Spring Boot и Bootstrap. Кроме того, проект зависит от корректной настройки и доступности выбранной базы данных и файлового хранилища.

3 Системные требования

3.1 Функциональные требования

Система должна предоставлять функционал для регистрации нового пользователя, требуя ввода email, пароля, ФИО и номера телефона, а также обеспечивать аутентификацию существующих пользователей через форму входа. Все защищенные страницы, такие как главная и личный кабинет, должны быть доступны только после успешного входа.

Аутентифицированный пользователь должен иметь возможность создавать, просматривать, редактировать и удалять свои собственные объявления о товарах. Создание объявления должно включать ввод названия, описания, цены и загрузку одной или нескольких фотографий товара.

Система должна отображать на главной странице каталог всех активных товаров в виде карточек и предоставлять страницу с детальной информацией о каждом товаре при клике на него. Также система должна обеспечивать возможность поиска товаров по ключевым словам, содержащимся в их названии или описании.

Аутентифицированный пользователь должен иметь возможность добавлять любой товар в свой персональный раздел "Понравившееся" и удалять его оттуда. Система должна отображать полный список этих товаров в соответствующем разделе личного кабинета.

Пользователь должен иметь возможность просматривать и редактировать информацию в своем профиле, а именно: ФИО, номер телефона и адрес электронной почты.

3.2 Нефункциональные требования

3.2.1 АТРИБУТЫ КАЧЕСТВА

Для системы критически важны несколько атрибутов качества. Удобство использования является paramount, так как оно напрямую влияет на удержание пользователей; его можно измерить тем, что новый пользователь должен интуитивно понять интерфейс и создать свое первое объявление без помощи сторонних инструкций менее чем за 5 минут. Производительность системы напрямую сказывается на пользовательском опыте; время загрузки главной страницы не должно превышать 2 секунд, а выполнение поискового запроса должно занимать не более 1 секунды. Надежность обеспечивает доверие к платформе и должна измеряться временем бесперебойной работы на уровне не менее 99.5%, а также способностью системы корректно обрабатывать ошибки ввода без полного отказа в работе. Безопасность является обязательным требованием для защиты пользовательских данных; пароли должны храниться только в хешированном виде с использованием стойких алгоритмов, а все запросы к защищенным ресурсам должны проверяться механизмами авторизации, с защитой от распространенных уязвимостей, таких как SQL-инъекции и XSS. Масштабируемость архитектуры важна для будущего роста и должна позволять наращивать вычислительные мощности сервера и пропускную способность базы данных без фундаментальных изменений в коде.