|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н. Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н. Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

ОТЧЕТ ПО ДОМАШНЕМУ ЗАДАНИЮ

Тема: «Онлайн-магазин продуктов ВКО «Алмаз-Антей»»

Студент: Макеев В.А.

Группа ИУ5Ц-74Б

Преподаватель: Канев А.И.

*Москва, 2022г.*

# **Введение**

Цель: разработать приложение онлайн-магазина продуктов ВКО «Алмаз-Антей».

Задачи:

* Разработать базу данных для сервиса;
* Реализовать интерфейс REST API, посредством которого будет происходить общение с базой данных;
* Создать пользовательский интерфейс выбора товаров;
* Организовать составление корзины и заказов;
* Реализовать на сайте функции для специального пользователя – менеджера: изменение статусов заказов, изменение и добавление товаров;
* Создать мобильное приложение нашего сайта.

Актуальность: Актуальность: понятный любому пользователю, удобный и минималистичный интерфейс призван упростить работу пользователя с приложением, а адаптивная верстка позволит также воспользоваться им и на других устройствах – смартфонах, планшетах и т.д., что позволит пользователю испытать лучший опыт пользования приложением. Алмаз-Антей – российский концерн, объединяющий предприятия, разрабатывающие и выпускающие вооружения для противовоздушной обороны и противоракетной обороны, а также гражданскую продукцию, связанную с военными разработками из систем управления, топопривязки и радиолокации. Приоритетными отраслями также являются: медицинская техника, связь, транспорт, телекоммуникационное оборудование и другие. В настоящее время, по понятным причинам, необходимо наладить контакт между различными структурами и органами, так или иначе вовлеченными в перечисленные выше области, для чего и служит наше приложение. Оно позволяет быстро находить необходимую продукцию, вести учет приобретений и сделок.

# **Бизнес-процесс.**

В нашей системе существует четыре типа пользователей: гость, авторизованный пользователь, менеджер и администратор системы.

Гость обладает возможностью просмотра списка всех товаров и страниц с описанием отдельных товаров, но не способен оформить заказ, добавить какую-либо товар в корзину.

У авторизованного пользователя появляется возможность оформлять заказы, формировать и изменять свою корзину, оплачивать заказы, а также просматривать полную историю своих заказов.

Менеджер, помимо прав авторизованного пользователя, получает доступ, к редактированию товаров, редактированию списка заказов пользователей – проверка заказа и установка различных статусов.

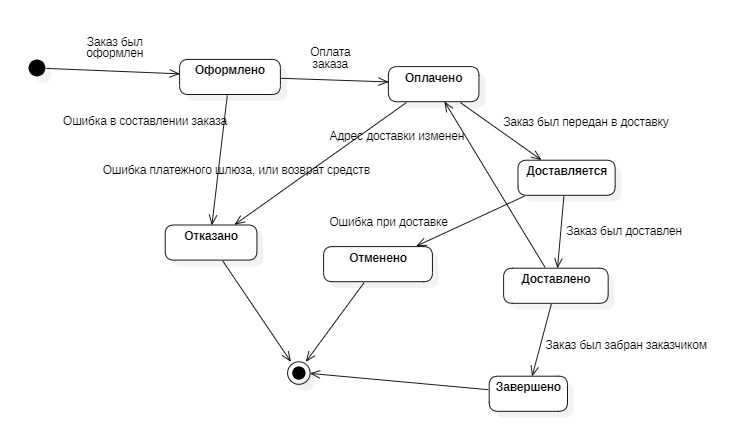
Администратор способен изменять данные в базе и управлять пользователями посредством соответствующей панели администратора.

Каждый заказ имеет свой жизненный цикл, начиная с формирования его корзины и до отклонения или доставки клиенту.

# **Бизнес-процесс. Диаграмма состояний**

Бизнес-процесс поддерживает определённую последовательность состояний заказа. Cначала, при оформлении заказа выдается статус «Оформлено» далее клиент переходит в платежный шлюз и оплачивает покупку. После подтверждения оплаты от шлюза, заказу выдают статус «Оплачено», иначе, если имеется ошибка или оплата не пришла, то выдается статус «Отказано». Далее если были допущены какие-то ошибки, то статус снова меняется на «Отказано».   
 Далее менеджер передает заказ в доставку, и заказ обновляет статус на «Доставляется». После если в ходе доставки, заказчик отменяет заказ, то статусу присваивается значение «Отменено». Если же доставка прошла успешно, то заказ становится «Доставленным» и будет ожидать покупателя. Если заказчик изменит адрес доставки, то он вернется назад в статус «Оплачено». Затем после подтверждения получения заказа, менеджер устанавливает статус «Завершено». Такие статусы как «Отказано», «Отменено» и «Завершено» являются конечными и ведут к завершению жизненного цикла заказа.

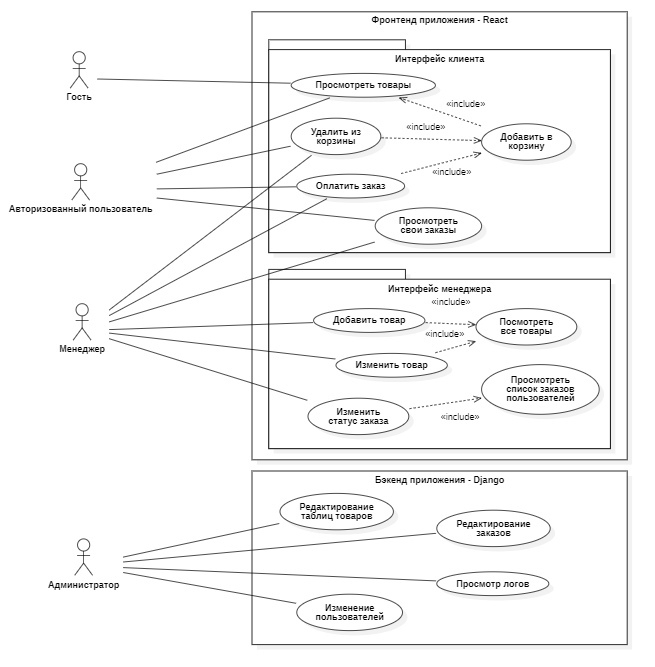
Все статусы заказов можно отследить на отдельной специальной странице.



# **Бизнес-процесс. Диаграмма прецедентов**

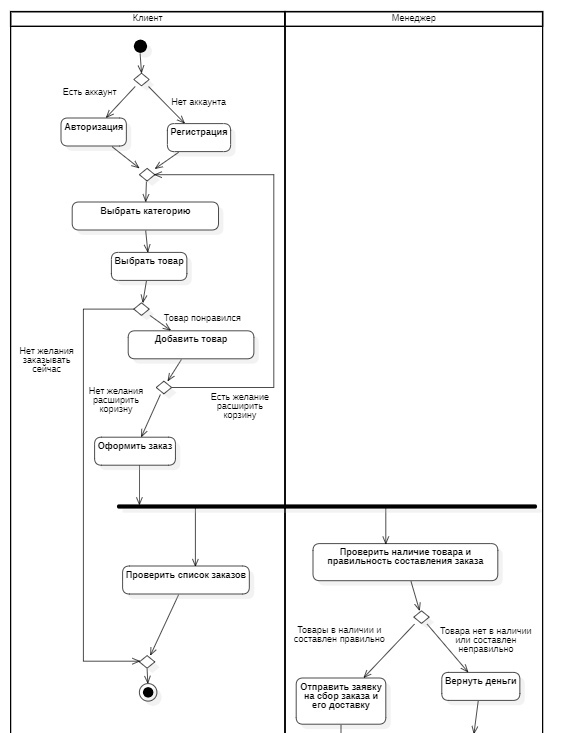
Гость: авторизация и регистрация, просмотр товаров.

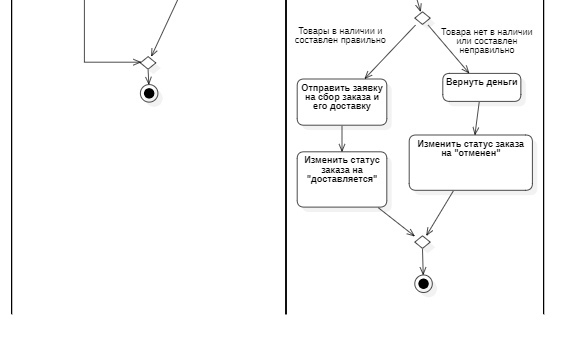
Авторизированный пользователь: просмотр товаров, добавление товаров в корзину, удаление товаров из корзины, оформление и оплата заказа, просмотр истории заказов.  
Менеджер: функции авторизованного пользователя и просмотр пользовательских заказов, изменение статусов заказов, и их удаление, добавление и редактирование данных в таблице товаров.  
Администратор: функции менеджера и создание, удаление или редактирование пользователей.



# **Диаграмма деятельности**

На диаграмме деятельности отображён функционал клиента – оформление заказа. Сначала пользователю необходимо авторизоваться. Если же он еще не зарегистрирован, то он может сначала зарегистрироваться. После этого клиента перебросит на страницу с выбором категории. После выбора категории пользователь может изучить соответствующие товары и выбрать нужный. Затем он может добавить товар в корзину, после чего может вернуться к поиску новых товаров или же прекратить поиск и перейти к корзине, где он может закончить её оформление или же отложить этот процесс. После оформления заказа, он производит оплату, после чего бизнес-процесс на стороне пользователя кончается. В это же самое время у менеджера появляется возможность начать взаимодействие с заказом. Он может оформить заявку на доставку и изменить статус на «Доставляется», либо же отменить заказ, если он был составлен неправильно или же есть нехватка товаров на складе. После этого их возможности кончаются.





# **Архитектура приложения**

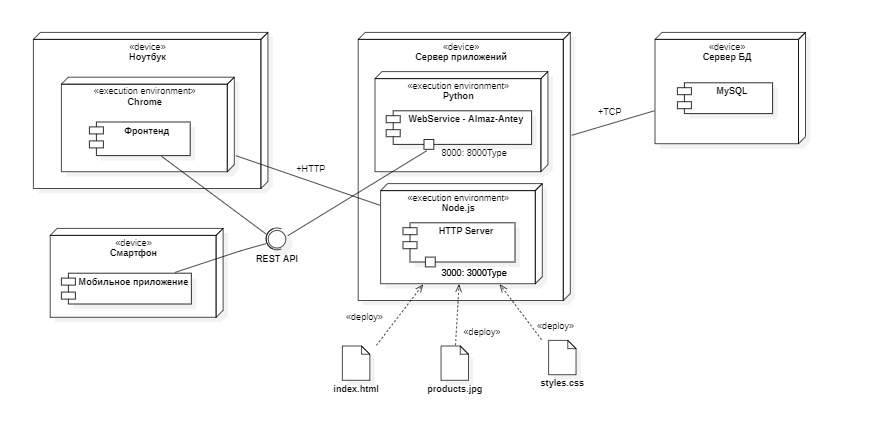
Приложение состоит из БД MySQL, HTTP сервера, веб-сервиса и фронтенда. Веб-сервис реализован на Django при помощи Django-rest-framework. Фронт написан на React на языке JS. Каждый из компонентов развернут в своей исполняемой среде.

# **Диаграмма развертывания**

Архитектура приложения отображена на диаграмме развёртывания, приведённой ниже.

На диаграмме изображены два пользовательских узла: ноутбук и телефон. На ноутбуке в браузере Chrome развёрнут фронтенд нашего сервиса, а на телефоне – мобильное приложение. Эти компоненты взаимодействуют с HTTP сервером, развернутом на 3000 порте и исполняемым средой Node.js, по http протоколу и получают от него статические файлы – одну html страницу нашего приложения, css стили, скрипты и картинки. Основную же информацию они получают от веб-сервиса, развернутого на 8000 порте и исполняемого средой Python, и реализующего REST API интерфейс.

По протоколу TCP сервер приложений взаимодействует с сервером базы данных, где используется СУБД MySQL.

****

# **ER-диаграмма**

На ER-диаграмме отображена структура базы данных, используемой в проекте. Опишем семантику базы данных: products – производимые продукты, group\_product – группа товаров, category\_product – категория товаров, cart – корзина, order – заказ, statuses – статусы заказа, ordercart – состав заказа, user – пользователь (клиент).

Рассмотрим поля, которые содержатся в таблицах:

Таблица products: id, name, price, description, amount, picture, is\_shown, category, group;

Таблица group\_product: id, name;

Таблица category\_product: id, name;

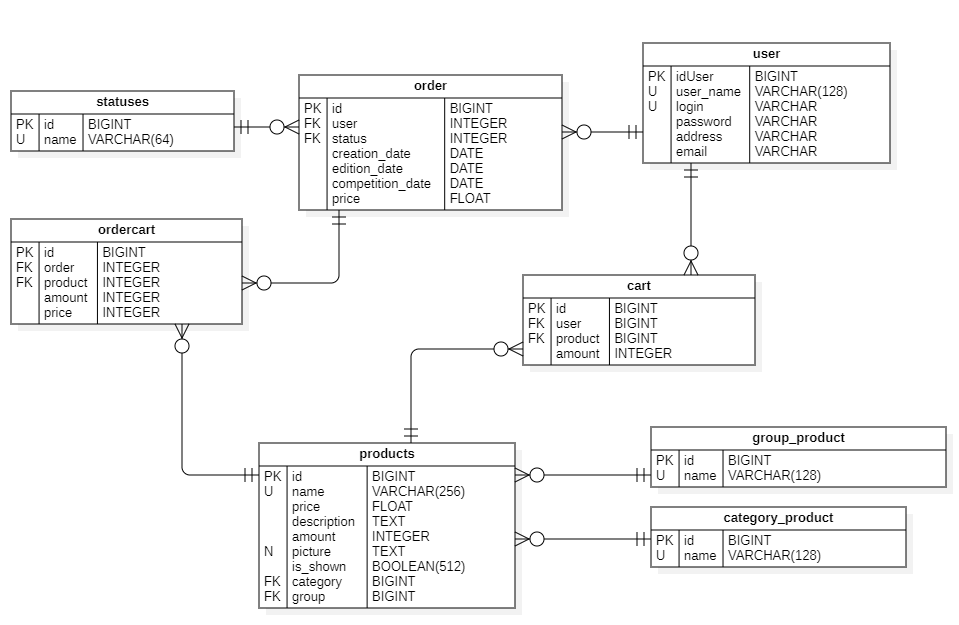
Таблица cart: id, user, product, amount;

Таблица order: id, user, status, creation\_date, edition\_date, competition\_date, price;

Таблица statuses: id, name;

Таблица ordercart: id, order, product, amount, price;

Таблица user: idUser, user\_name, login, password, address, email.



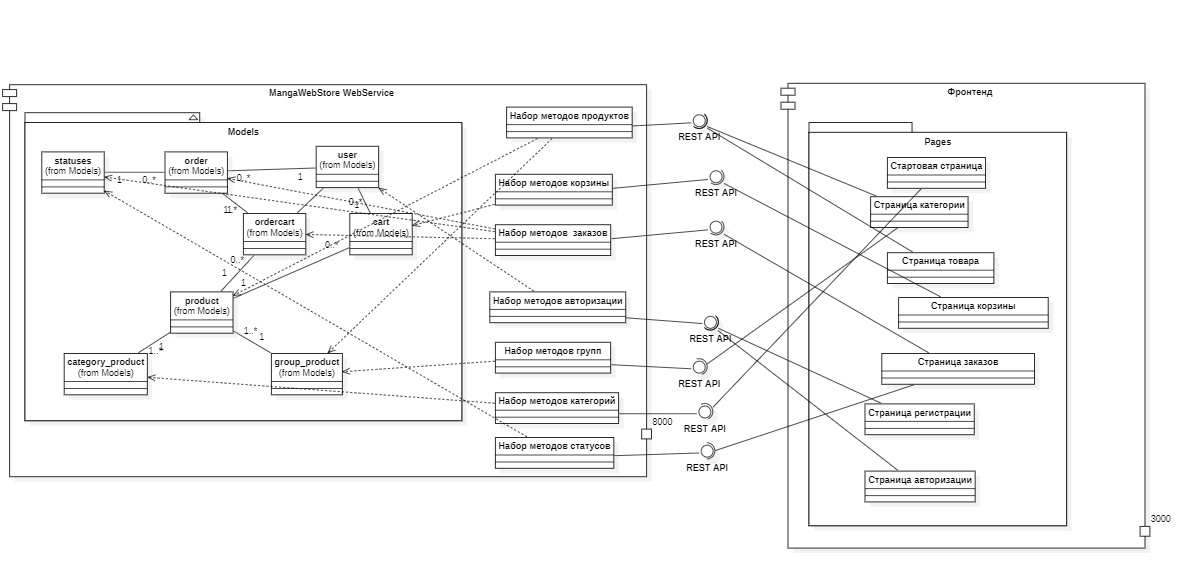
# **Диаграмма компонентов**

Ниже приведена диаграмма, описывающая компонентную структуру проекта. На ней отображена фронтенд и бэкенд нашей системы.

Фронтенд включает в себя множество страниц, асинхронные запросы к бэкенду и методы регистрации и авторизации. Регистрация и авторизация предоставлены пользователям соответствующими страницами и используют методы Django на бэкенде.

Остальные страницы, содержащие или предоставляющие возможность редактирования информации используют REST API интерфейс, реализованный соответствующими методами бэкенда.

Далее рассмотрим компонентную структуру нашего веб-сервиса. Для авторизации и регистрации пользователей используется схема пользователей Django и технология JWT-токенов. Остальные методы предоставляют в определенном виде информацию из модели, которые связывают бэкенд и БД MySQL.

****

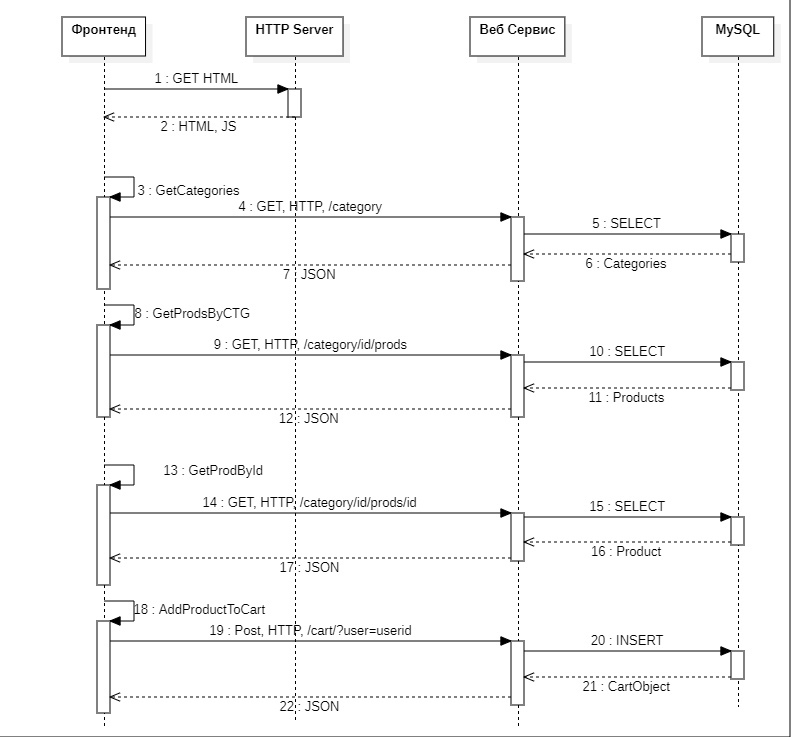
# **Диаграмма последовательности**

Диаграмма последовательности, приведённая ниже, описывает, как осуществляются запросы в приложении.

Сначала фронтенд отправляет http get запрос HTTP серверу на получение главной страницы нашего приложения. В ответ он получает эту страницу и некоторые статические файлы – иконки, скрипты и прочее.

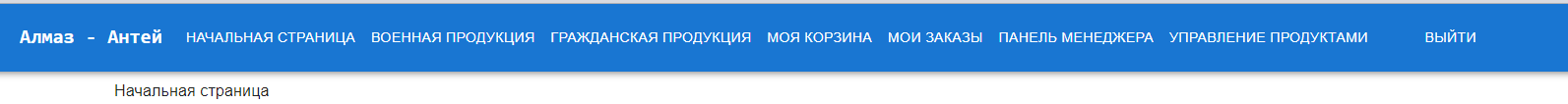
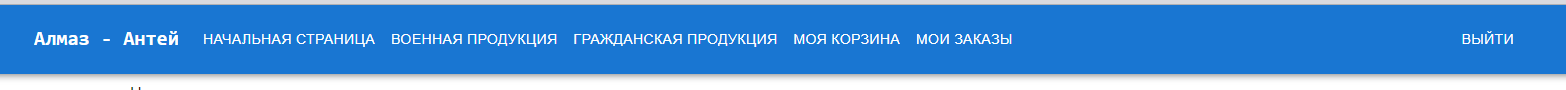
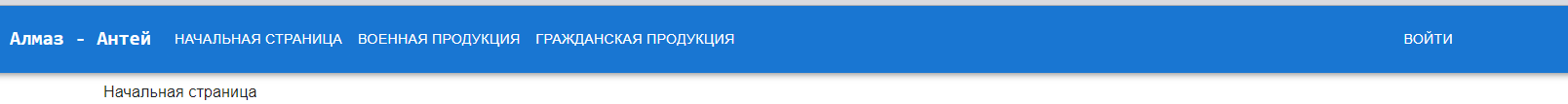
Далее с фронтенда могут отправляются http запросы на получение данных из моделей, и веб-сервис на бэкенде, обрабатывая эти запросы, обращается к базе данных и, считывая из неё записи, передаёт их в виде ответа фронтенду.

В приложении поддерживается регистрация пользователя.

****

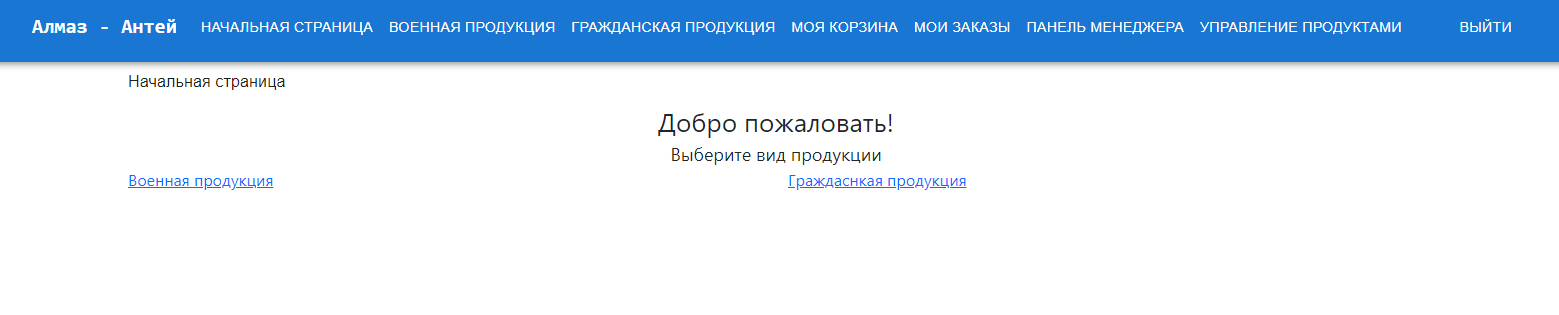
# **Описание интерфейса**

На каждой странице приложения доступно меню приложения в шапке страницы, где представлены переходы на начальную страницу и страницу с военной продукцией или гражданской продукцией. Если гость авторизуется, у него появится 2 новых опции: корзина и список заказов. Если же авторизованный пользователь является менеджером, то у него также помимо опций авторизованного пользователя есть опции выбора страниц изменения товаров и статусов заказов. Также есть кнопка для произведения авторизации и выхода из аккаунта.



Главная страница:

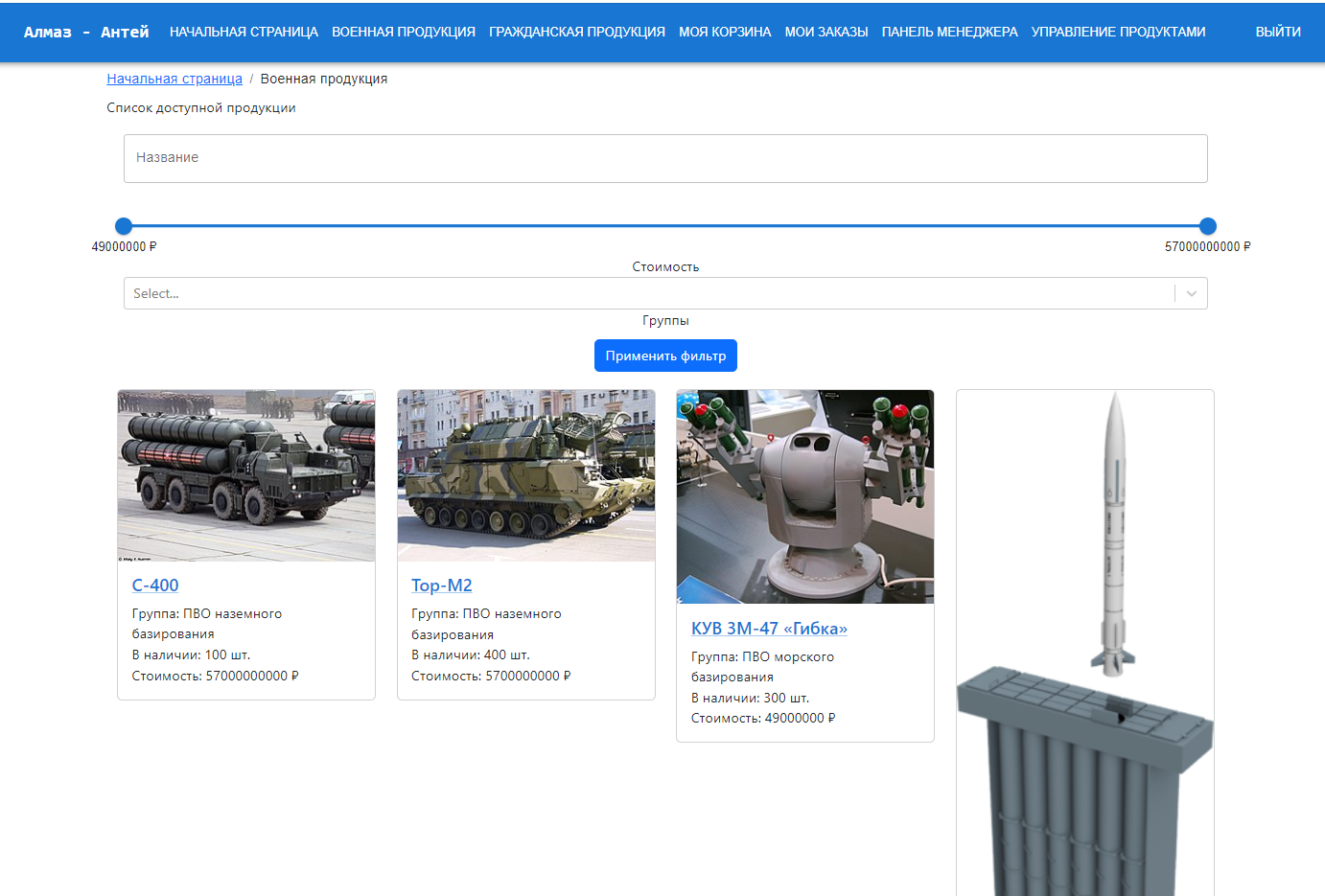
Главная страница представляет собой страницу с переходами по категориям и меню сайта в шапке страницы. Также есть навигационная панель, указывающая, где сейчас находится пользователь.



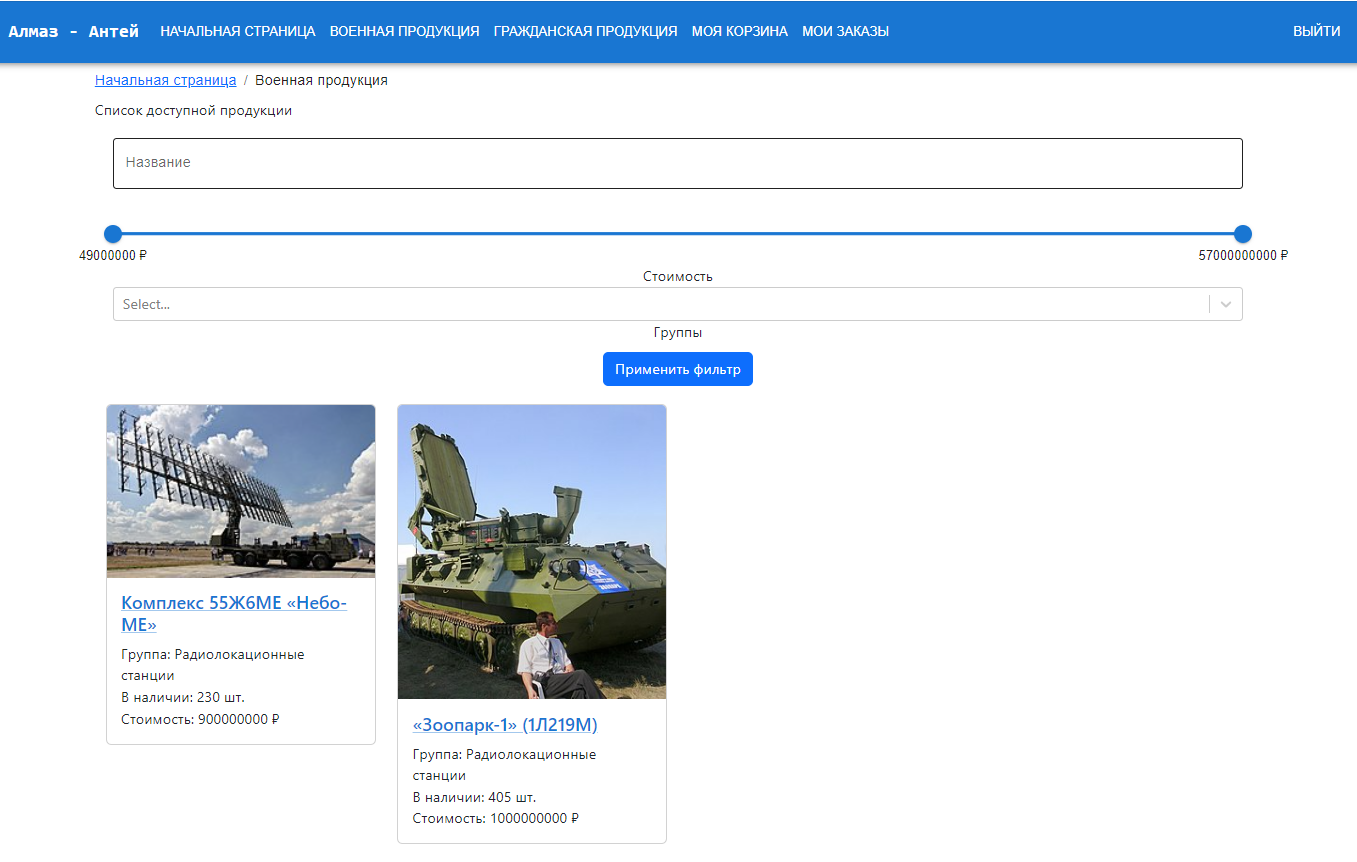
Список товаров категории:

Эта страница содержит товары определенной категории, они представлены в виде карточек, нажатие на которые перенаправит вас на страницу самого товара. Сверху страницы располагается навигационная панель, на которой можно перейти на предыдущие страницы в иерархии. Также на странице располагается блок фильтров для поиска определенных товаров по таким полям как: цена, группа товаров и название. Фильтрация осуществляется после нажатия на кнопку «Применить фильтр».

Пример страницы

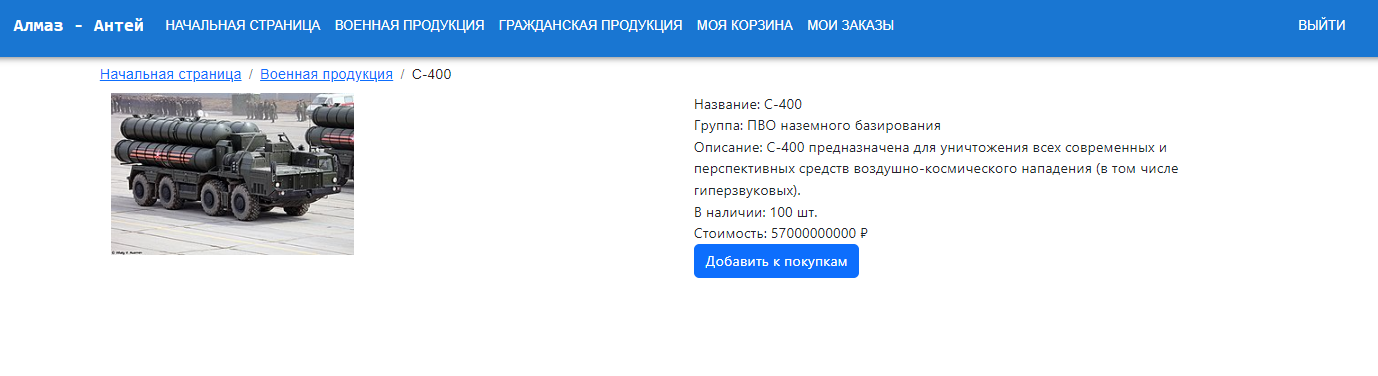


Фильтрация



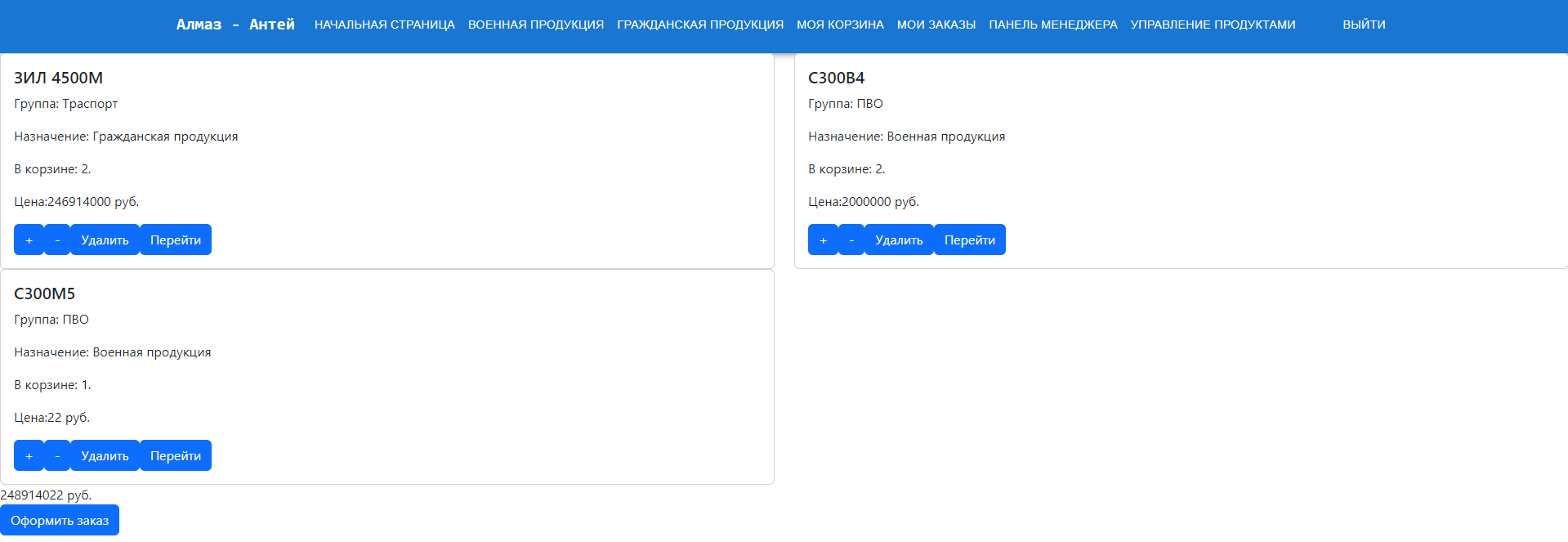
Страница отдельного товара:

Перейдя по изображения на странице можно увидеть подробную информацию о товаре, а также добавить его в корзину, если пользователь авторизован.



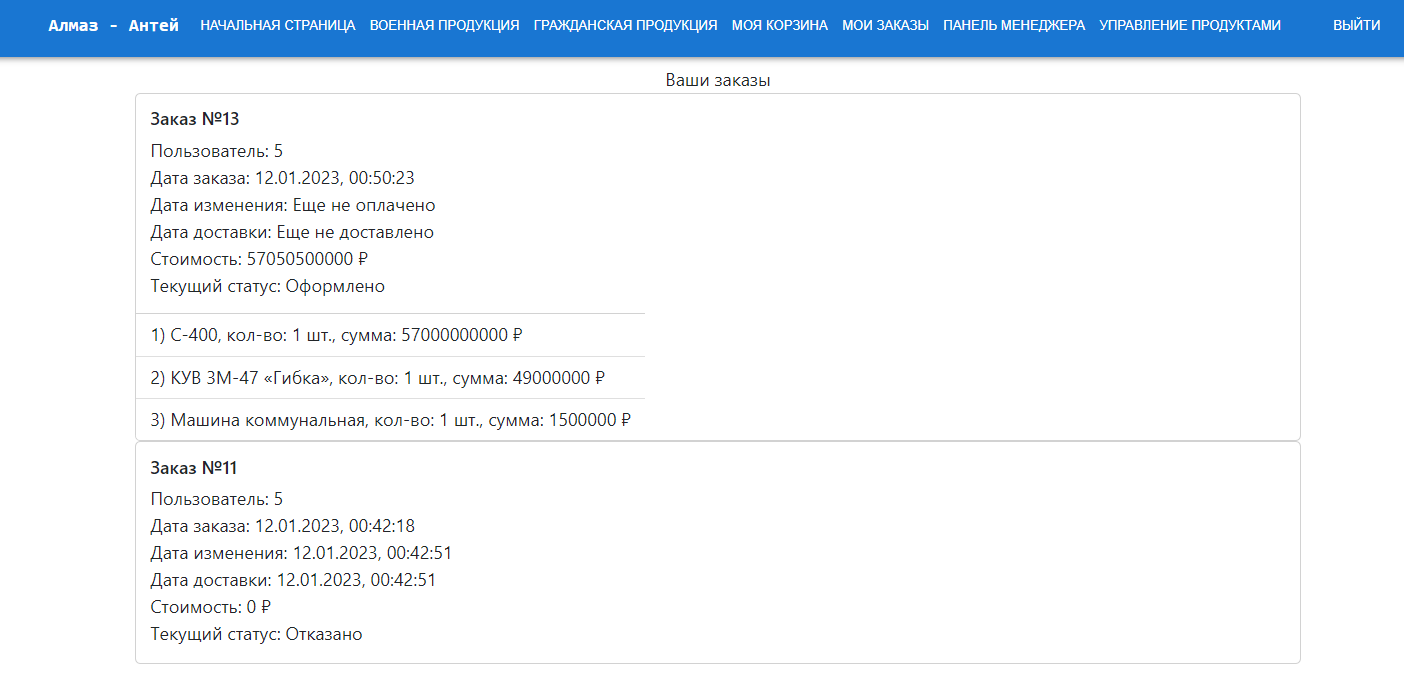
Страница корзины:

Здесь можно увидеть весь список товаров, которые были добавлены в корзину. На каждой карточке товара есть информация о итоговой сумме товара, его количестве и несколько кнопок: «+» увеличивает количество товара в корзине на 1, «-» уменьшает количество на один, вплоть до удаления товара из корзины, «Удалить» удаляет товар из корзины, а «Перейти» перенаправляет на страницу товара. Кнопка «Оформить товар» собирает корзину в заказ, очищая корзину после этого. Все заказы можно посмотреть на странице заказов.



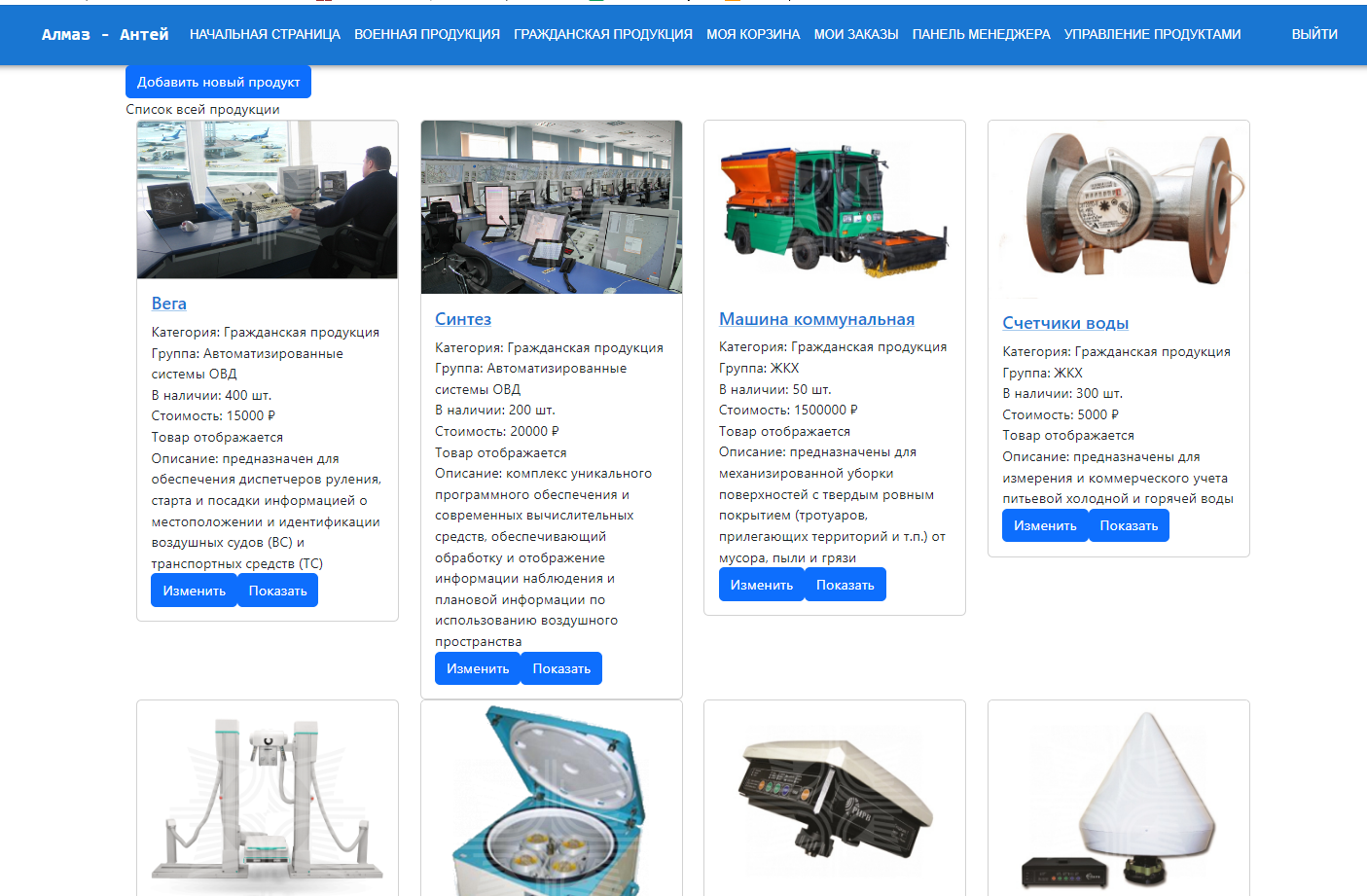
Страница мои заказы:

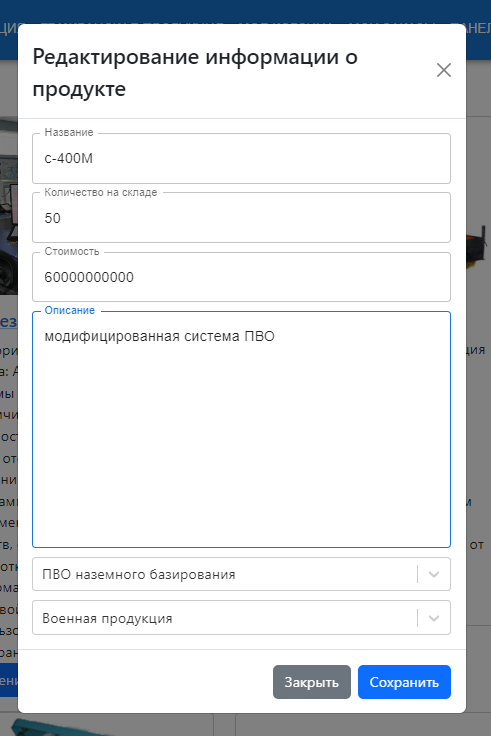
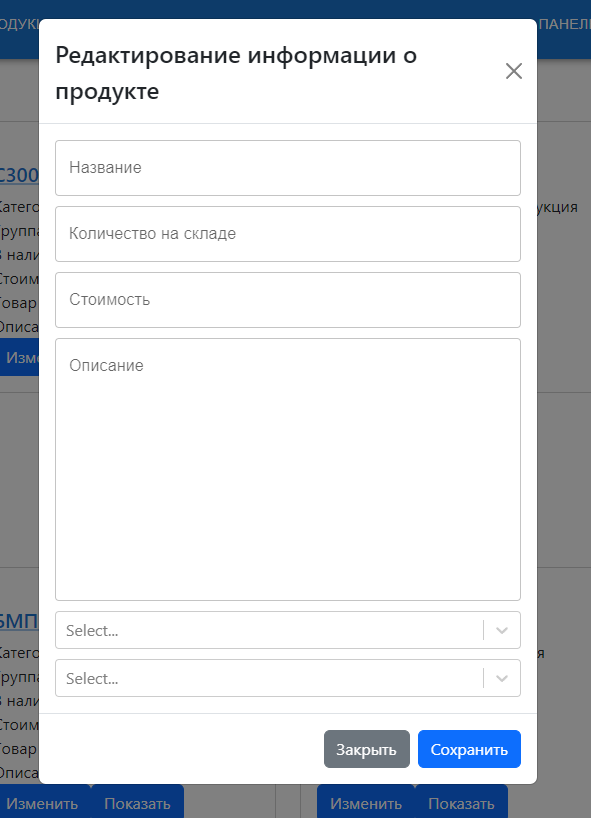
На данной страницы доступен просмотр всех заказов пользователей, в том числе уже доставленных и находящихся в процессе в данный момент. Карточка каждого заказа содержит информацию о его номере в системе, дате создания заказа, изменения заказа и завершения заказа, итоговой стоимости, текущем статусе и списке товаров, составляющих в заказ. У менеджера также отображается имя клиента, которому принадлежит заказ.



Страница менеджера по изменению списка товаров:

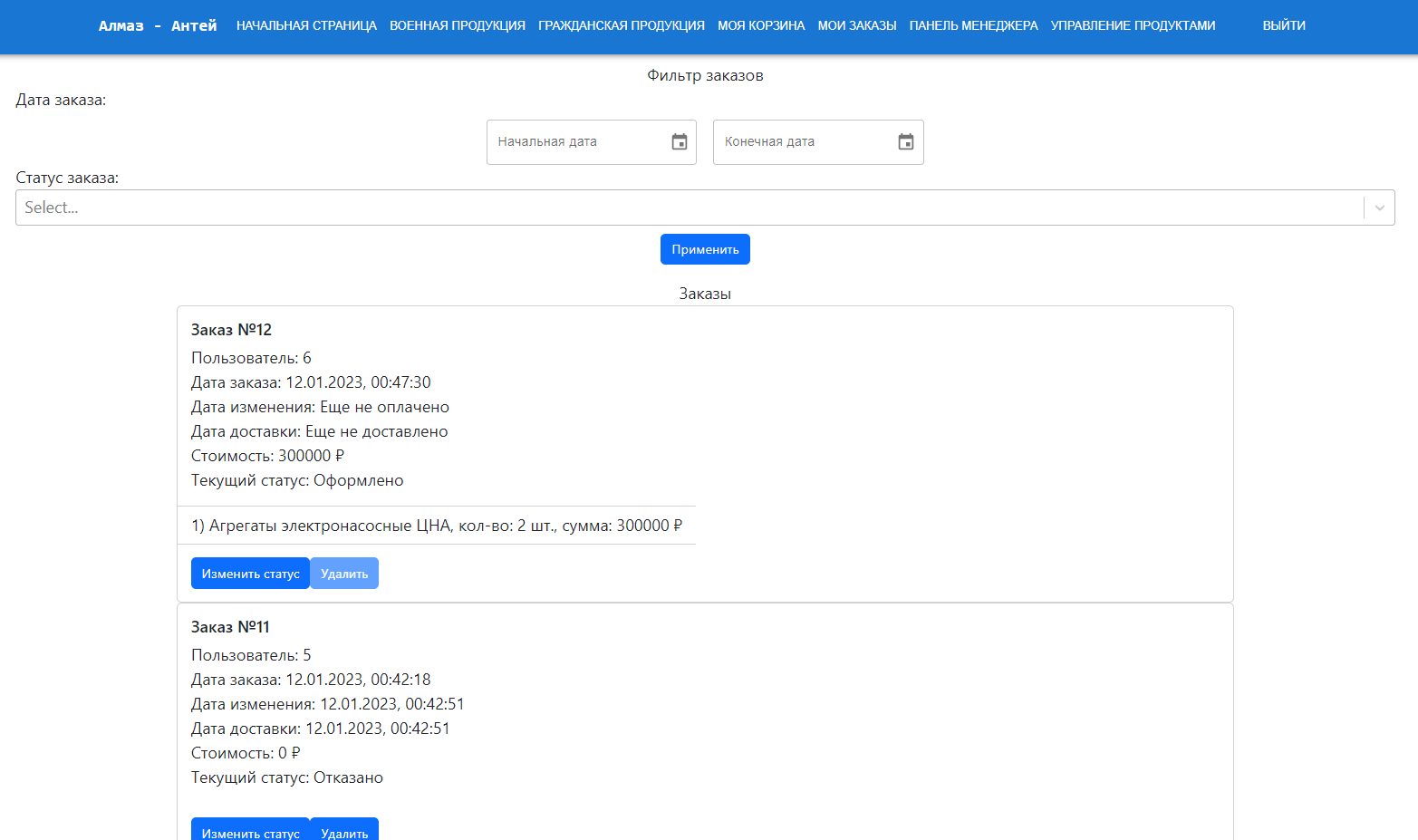
Здесь есть карточки товаров, со всех информацией о них, а также с кнопкой добавления нового товара. Нажатие на кнопку «Добавить новый товар» вызовет модальное окно, куда нужно вписать необходимую информацию, после чего нажать «Сохранить». На карточке товара есть кнопка изменить, которая откроет схожее модальное окно, но в котором данные уже будут заполнены. Также кнопка «Скрыть» или «Показать» изменит статус показа товара на соответствующий.



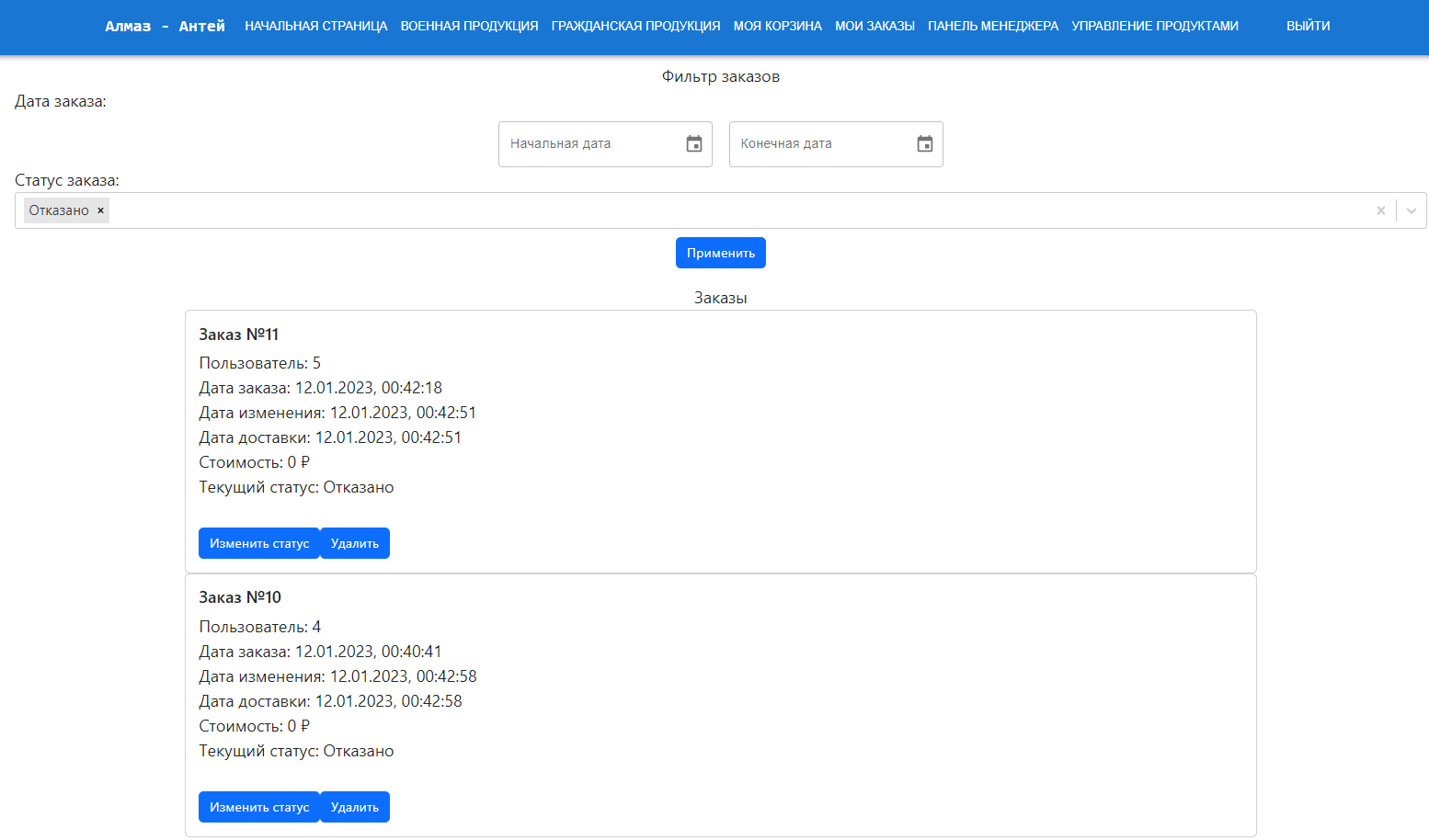


Страница менеджера по изменению статусов заказов:

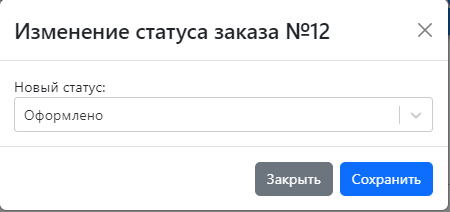
Эта страница похожа на страницу списка заказов конкретного пользователя, за исключением того, что здесь отображаются все заказы, добавлен фильтр по статусу и дате заказа, отображаются имена клиентов.



Демонстрация фильтрации:

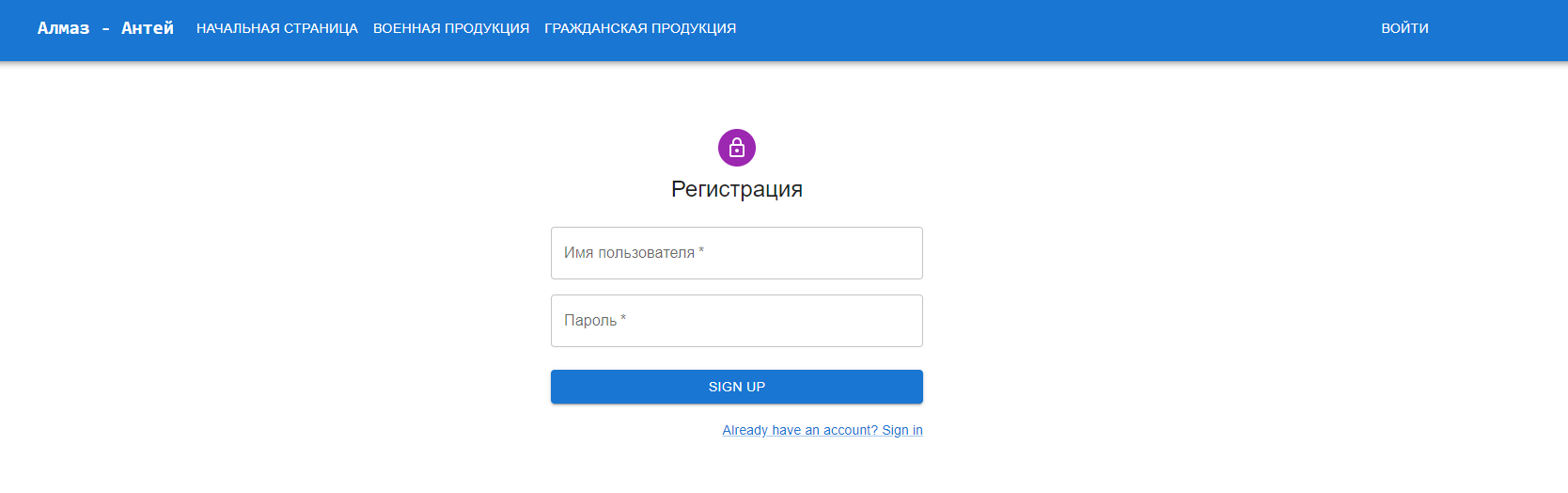


Изменение статусов также реализовано через модальное окно, в котором из списка можно выбрать новый статус.



Страница регистрации:

Содержит форму с двумя полями: логин и пароль, которые являются обязательными для заполнения. По кнопке «Sign Up» создается новый пользователь в системе, после чего вы переходите на страницу для авторизации. Если же пользователь, уже имеющий аккаунт, оказался на этой странице, он может по соответствующей ссылке перейти на страницу авторизации.



Страница авторизации:

Похожа на страницу регистрации, но содержит только поля логина и пароля, после правильного ввода которых, будет произведен вход в аккаунт.

