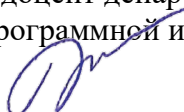


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

СОГЛАСОВАНО
Руководитель,
доцент департамента
Программной инженерии


Г. Н. Жукова
«12» 05 2022 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»


В. В. Шилов
«13» мая 2022 г.

ВЕБ-ПРИЛОЖЕНИЕ “NOTIPRICE”


Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.03-01 ПЗ 01-1-ЛУ

Исполнитель

студент группы БПИ198

 /Бакытбек уулу Н. /

«9» мая 2022 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.04.03-01 ПЗ 01-1-ЛУ

Москва 2022

УТВЕРЖДЕН
RU.17701729.04.03-01 01-1

ВЕБ-ПРИЛОЖЕНИЕ “NOTIPRICE”

Пояснительная записка

RU.17701729.04.03-01 ПЗ 01-1

Листов 44

<i>Инв. № подл</i>	<i>Подп. и дата</i>	<i>Взам. инв. №</i>	<i>Инв. № дубл.</i>	<i>Подп. и дата</i>
RU.17701729.04.03-01 ТЗ 01-1-ЛУ				

Москва 2022

Оглавление

1	Введение.....	4
1.1	Наименование программы:.....	4
1.2	Документы, на основании которых ведется разработка:.....	4
2	Назначение и область применения.	5
2.1	Назначение программы.....	5
2.1.1	Функциональное назначение.....	5
2.1.2	Эксплуатационное назначение.....	5
2.2	Краткая характеристика области применения.....	5
3	Технические характеристики.	6
3.1	Постановка задачи на разработку программы.....	6
3.2	Описание функционирования программы.....	6
3.2.1	Описание функционирования различных окон приложения	6
3.2.2	Описание структуры базы данных PostgreSQL	13
3.2.3	Диаграмма классов – заголовок с номером.....	15
3.3	Описание и обоснование выбора метода организации входных и выходных данных 15	
3.3.1	Описание метода организации входных и выходных данных.....	15
3.3.2	Обоснование выбора метода организации входных и выходных данных.....	15
3.4	Описание и обоснование выбора состава технических и программных средств	16
3.4.1	Состав технических и программных средств	16
4	Ожидаемые технико-экономические показатели	17
4.1	Предполагаемая потребность.....	17
1.1	Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами	17
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	18
	ПРИЛОЖЕНИЕ 2 ОПИСАНИЕ КЛАССОВ	19
	ПРИЛОЖЕНИЕ 3 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ	21

ПРИЛОЖЕНИЕ 4	43
ТЕРМИНОЛОГИЯ	43
Лист регистрации изменений	44

1 Введение.

1.1 Наименование программы:

Веб-приложение “NotiPrice” (“NotiPrice” web application).

1.2 Документы, на основании которых ведется разработка:

Основанием для разработки настоящего продукта является учебный план, бакалавриата 09.03.04 по направлению "Программная инженерия", а тема данного курсового проекта, утвержденная академическим руководителем направления.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2 Назначение и область применения.

2.1 Назначение программы

2.1.1 Функциональное назначение

В данной программе можно отслеживать цены на товары в интернет-магазинах и получать уведомление, при заданных пользователем условиях изменения цены.

2.1.2 Эксплуатационное назначение

Настоящая программа предназначена для всех людей, которые совершают покупку в интернет-магазинах. Самим интернет-магазинам предоставлять данный функционал невыгодно, потому что многие покупатели будут ждать пока цены снизятся.

2.2 Краткая характеристика области применения

Данная программа полезной будет для всех, кто совершает покупки в веб-приложениях. У большинства людей бывают такие случаи, когда они заранее знают, что хотят купить. Программа помогает отслеживать цены на товары.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3 Технические характеристики.

3.1 Постановка задачи на разработку программы

Настоящая программа должна содержать следующий функционал:

1. Регистрация нового пользователя или авторизация пользователя.
2. Добавление товара для отслеживания.
3. Редактирование товара для отслеживания.
4. Удаление товара для отслеживания.
5. Уведомление о изменении цены отслеживаемого товара.

3.2 Описание функционирования программы

3.2.1 Описание функционирования различных окон приложения

Каждая страница сайта редактируются в .js файлах фронтенд программы. Описание классов проекта можно найти в Приложение 2, 3. В процессе разработки настоящей программы были использовались ресурсы [1]-[5].

После старта данной программы пользователь может видит окно авторизации (рисунок 1), при условии того, что пользователь еще не выполнил вход.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

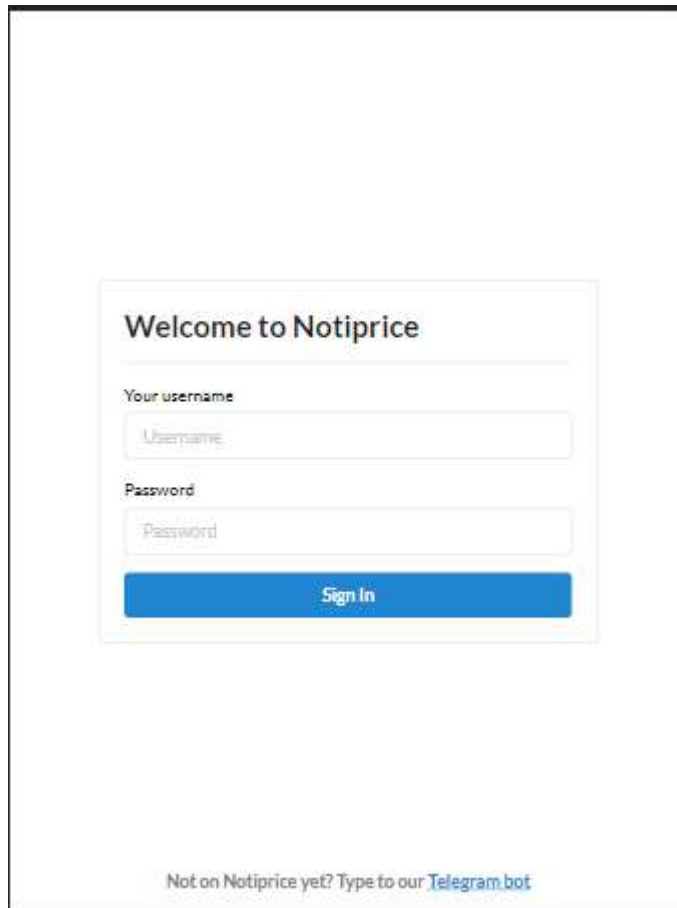


Рисунок 1 - Окно аутентификация. Аутентификация пользователей.

Окно аутентификации

Для аутентификации пользователей используется самостоятельная реализация при помощи Spring Security. Spring Security — это популярная и легко настраиваемая платформа аутентификации и контроля доступа. Это фактический стандарт для обеспечения безопасности приложений на основе Spring. Также для аутентификации запросов от фронтенда к бэкенду используется JWT. JWT— это стандарт, который определяет способ безопасной передачи информации между сторонами в виде объекта JSON. Эта информация может быть проверена и ей можно доверять, поскольку она подписана с использованием общего ключа с помощью алгоритма HS512. Также пароль хранится в зашифрованном виде. После аутентификации клиенту возвращается токен, который он будет использовать для получения доступа к остальным контроллерам.

В окне аутентификации содержится:

1. Поле для ввода имени пользователя.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. Поле для ввода пароля.
3. Кнопку для входа в приложение.
4. Кнопку для создания новой учетной записи.

При нажатии на кнопки создания учетной записи открывается ссылка на Телеграм бота.

После авторизации пользователь попадет в окно отслеживаемых товаров (рисунок 3).

Окно регистрации

В Телеграм боте пользователь вводит команду “/signup” и получает ссылку на регистрацию с идентификатором чата. Добавление пользователя в базу данных происходит через класс AuthController. Перед добавлением пароль хешируется алгоритмом BCrypt.

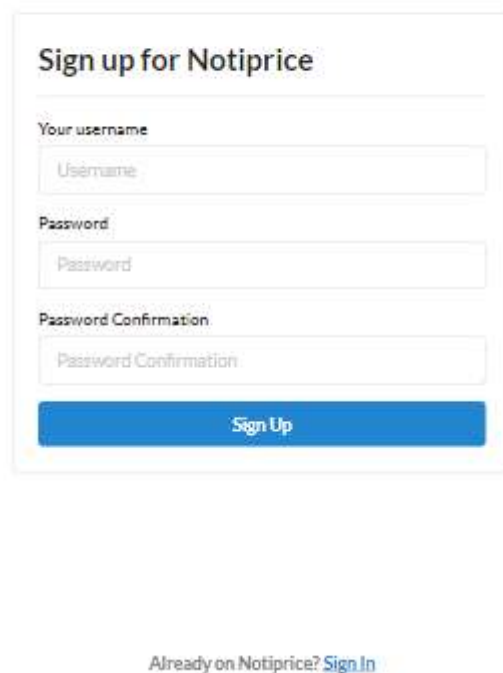


Рисунок 2 - Окно регистрации. Регистрация пользователей.

В окне регистрации содержится:

1. Поле для ввода имени пользователя.
2. Поле для ввода пароля.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. Поле для повторного ввода пароля.
4. Кнопку для регистрации.
5. Кнопку для перехода в страницу аутентификации.

После отправки необходимых данных бэкенд сохраняет нового пользователя, а страница автоматически переходит в окно аутентификации (рисунок 1).

Окно отслеживаемых продуктов

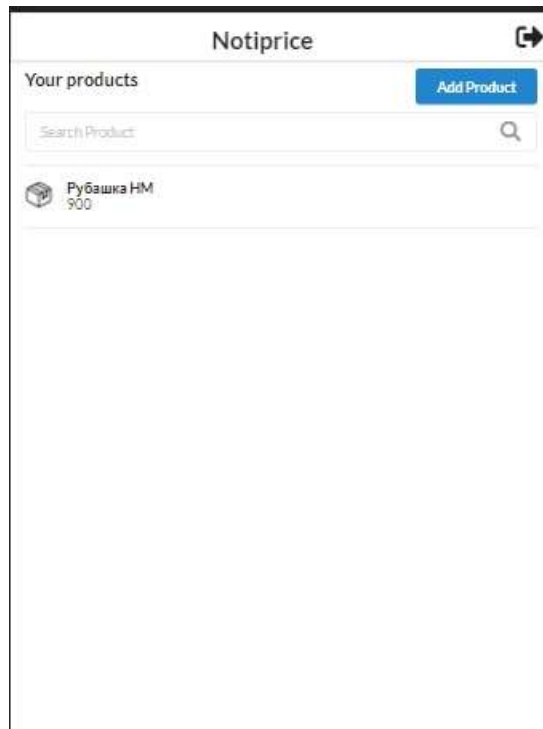


Рисунок 3 - Окно отслеживаемых товаров. Отображаются товары, за которыми следит пользователь.

Фронтенд для управления товарами отправляет запросы на класс ProductController. Класс PriceScanner запускает сканирование с определенным интервалом для проверки изменения цен, он проверяет товары, которые не проверялись определенный интервал времени, если их цена меняется, то пользователям отправляются сообщения через Телеграм бот.

На данном окне пользователь может увидеть список своих отслеживаемых товаров. В странице содержатся:

- 1) Кнопка для выхода из программы, которая удаляет информацию о пользователе из браузера.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 2) Кнопка для добавления нового товара.
- 3) Поле для поиска товара.
- 4) Список отслеживаемых товаров. По нажатию происходит переход в окно отслеживаемого товара (рисунок 4).

Окно отслеживаемого товара



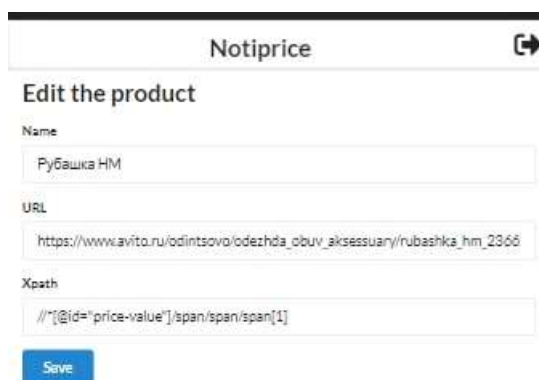
Рисунок 4 - Окно отслеживаемого товара. Отображается информация о товаре.

После нажатия на название продукта в списке отслеживаемых продуктов пользователь попадает в окно отслеживания товара. В окне содержатся:

- 1) Название товара.
- 2) Ссылка на товар в интернет-ресурсе.
- 3) Кнопка для удаления товара.
- 4) Кнопка для редактирования данных о товаре (рисунок 5).
- 5) Кнопка для возврата на страницу отслеживаемых товаров (рисунок 3).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Окно редактирования отслеживаемого товара



Notiprice

Edit the product

Name

Рубашка HM

URL

https://www.avito.ru/odintsovo/odezhda_obuv_aksessuary/rubashka_hm_2366

Xpath

//*[@id="price-value"]/span/span/span[1]

Save

Рисунок 5 - Окно редактирования товара. Редактируется информация о товаре.

В окне содержатся:

- 1) Поле для редактирования названия товара.
- 2) Поле для редактирования URL товара.
- 3) Поле для редактирования Xpath товара. Она используется для извлечения цены из сайта товара.
- 4) Кнопка для сохранения изменений.

Окно для добавления товара

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

The figure shows three sequential screenshots of the 'Notiprice' application interface for adding a product:

- Step 1:** The 'Add a product' screen. The 'Name' field contains 'Вторая рубашка'. The 'URL' field contains 'https://www.avito.ru/odintsovo/odezhda_obuv_aksessuary/rubashka_hm_2366'. A 'Next' button is visible at the bottom.
- Step 2:** The 'Add a product' screen. The 'Xpath' field contains '//[@id="price-value"]/span/span/span[1]'. A 'Next' button is visible at the bottom.
- Step 3:** A confirmation screen titled 'Add a product'. It displays the instruction: 'Please check if the selection valid and confirm adding of the product.' Below this is a 'Confirm' button. At the bottom, there are three numbered instructions: '1) Click here', '2) Check if the price of the product is highlighted', and '3) Press 'Confirm' button'.

Рисунок 6 - Окно добавления товара. Добавление товара для отслеживания.

Окно содержит форму, которая состоит из трех шагов. В первом шаге пользователь вводит название и URL продукта, во втором шаге вводит Xpath цены для извлечения информации из сайта, если в программе уже было это значение и оно корректно, то пользователю предлагается автоматически это значение по умолчанию, а в третьем шаге пользователь проверяет корректность введенных данных и подтверждает добавление.

Окно состоит из:

- 1) Поле для ввода названия продукта.
- 2) Поле для ввода URL продукта.
- 3) Кнопка для перехода в следующий шаг.
- 4) Поле для ввода Xpath для цены продукта.
- 5) Кнопка для перехода на предыдущий шаг.
- 6) Кнопка для подтверждения добавления товара.
- 7) Ссылку для проверки корректности введенных данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.2.2 Описание структуры базы данных PostgreSQL

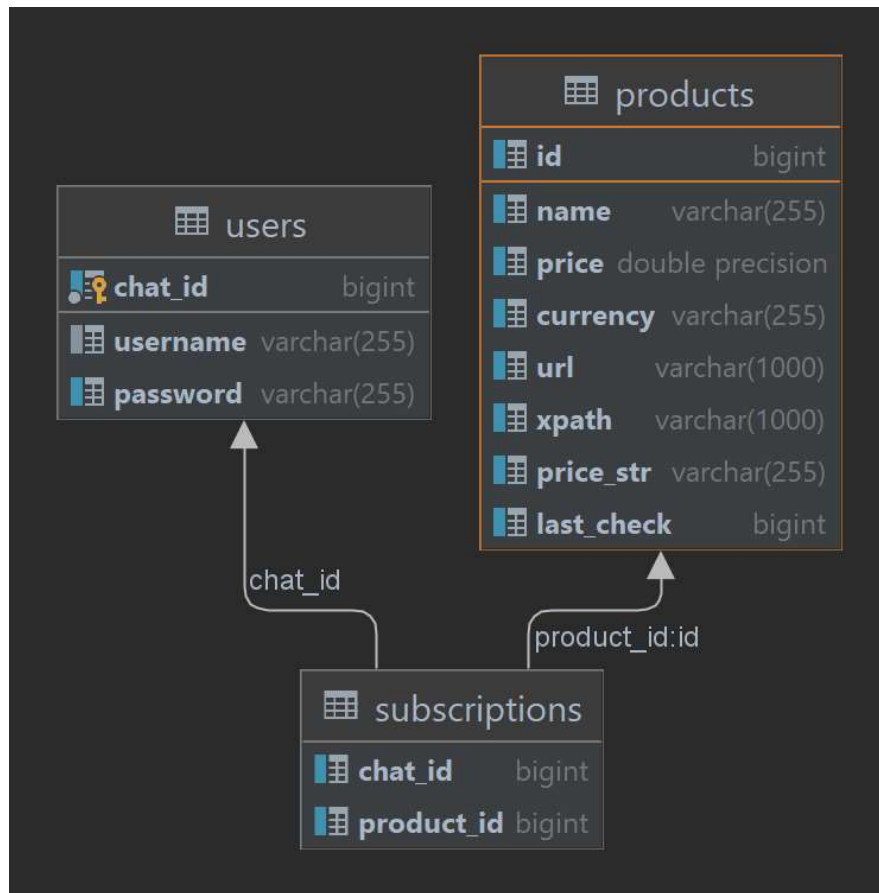


Рисунок 7 -Диаграмма базы данных.

Имеется три таблицы:

1. users -таблица для хранения данных о пользователях, отображен в бэкенде классом User (Приложение 1).

Название	Назначение
chat_id	Идентификатор чата пользователя в Телеграме. Первичный ключ.
username	Логин пользователя.
password	Пароль пользователя. Хранится в зашифрованном виде.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. products -таблица для хранения товаров, в бэкенде отображен классом Product (Приложение 1).

Название	Назначение
id	Идентификатор товара.
name	Название товара.
price	Цена товара в виде десятичного числа.
currency	Валюта.
url	URL на товар.
xpath	Путь до цены на странице товара.
price_str	Значение по пути цены товара на странице.
last_check	Время последней проверки цены.

3. subscriptions – таблица для хранения данных о подписке пользователя на товар, в бэкенде отображен классом Subscription (Приложение 1).

Название	Назначение
chat_id	Внешний ключ таблицы users.
Product_id	Внешний ключ таблицы products.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

The UML class diagram illustrates the architecture of the Notprice application. It features several classes and their interactions:

- CustomUserDetails** (Entity): Associated with **CustomUserDetailsService** (Service) via a 1-to-1 association.
- UserDao** (DAO): Associated with **User** (Entity) via a 1-to-1 association. It is also associated with **UserService** (Service) via a 1-to-1 association.
- SubscriptionDao** (DAO): Associated with **Subscription** (Entity) via a 1-to-1 association. It is also associated with **ProductService** (Service) via a 1-to-1 association.
- ProductDao** (DAO): Associated with **Product** (Entity) via a 1-to-1 association. It is also associated with **PriceScanner** (Service) via a 1-to-1 association.
- UserService** (Service): Associated with **UserDao** (DAO) and **User** (Entity). It is also associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- ProductService** (Service): Associated with **SubscriptionDao** (DAO) and **Subscription** (Entity). It is also associated with **ProductDao** (DAO) and **Product** (Entity).
- PriceScanner** (Service): Associated with **ProductDao** (DAO) and **Product** (Entity).
- ProductController** (Controller): Associated with **ProductService** (Service) and **PriceScanner** (Service).
- RestTemplateResponseErrorHandler** (Handler): Associated with **ProductController** (Controller).
- CustomUserDetailsService** (Service): Associated with **CustomUserDetails** (Entity) and **UserService** (Service).
- JwtProvider** (Service): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- JwtFilter** (Filter): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- AuthController** (Controller): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- UserController** (Controller): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- ServiceConfig** (Configuration): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- ControllerExceptionHandler** (Handler): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- NotpriceApplicationKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- CustomUserDetailsKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- NotpriceApplication** (Application): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- PriceScannerKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- ProductDtoKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- MyFirstBot** (Bot): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- UserDtoKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- JwtFilterKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- ProductKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- ScarperKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).
- UserKt** (Kotlin): Associated with **CustomUserDetailsService** (Service) and **CustomUserDetails** (Entity).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.4 Описание и обоснование выбора состава технических и программных средств

3.4.1 Состав технических и программных средств

В состав технических средств должно входить персональный компьютер с установленным интернет-браузером Яндекс версии 22.1.5.810 (64-bit):

1. Количество ядер - 1 и выше
2. Частота работы процессора – 1.3 ГГц и выше
3. Объем оперативной памяти - 2 ГБ и выше
4. Объем встроенной памяти - 32 ГБ и выше

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4 Ожидаемые технико-экономические показатели

4.1 Предполагаемая потребность

При помощи настоящего приложения пользователи могут сэкономить деньги, поэтому программа будет пользоваться спросом.

1.1 Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Преимуществами программы являются:

1. Является бесплатным.
2. Удобный интерфейс.
3. Не требует денежных затрат.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Начало работы [Электронный ресурс]// URL: <https://ru.reactjs.org/docs/getting-started.html/> (Дата обращения 14.3.2022, режим доступа: свободный).
2. Spring Boot Reference Documentation [Электронный ресурс]// URL: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> (Дата обращения 14.3.2022, режим доступа: свободный).
3. Telegram Bot API [Электронный ресурс]// URL: <https://core.telegram.org/bots/api> (Дата обращения 14.3.2022, режим доступа: свободный).
4. Кеера [Электронный ресурс]// URL: <https://keepa.com/#!> (Дата обращения 14.3.2022, режим доступа: свободный). <https://pricehistory.in/>
5. Price History [Электронный ресурс]// URL: <https://pricehistory.in/> (Дата обращения 14.3.2022, режим доступа: свободный).

ПРИЛОЖЕНИЕ 2
ОПИСАНИЕ КЛАССОВ

Таблица П1 - Описание и функциональное назначение классов

Класс	Назначение
ServiceConfig	Конфигурация для программы.
AuthController	Контроллер для аутентификации и регистрации.
ProductController	Контроллер отслеживаемых товаров.
UserController	Контроллер пользователей.
ProductDao	DAO продукта для работы с базой данных.
SubscriptionDao	DAO для класса Subscription для работы с базой данных.
UserDao	DAO для класса User для работы с базой данных.
ProductDto	DTO товара.
UserDto	DTO пользователя.
Product	Для отображения таблицы products из базы данных.
Subscription	Для отображения таблицы subscriptions из базы данных.
User	Для отображения таблицы users из базы данных.
ControllerExceptionHandler	Перехватывает исключение, если они возникают в результате вызова контроллера.
RestTemplateResponseErrorHandler	Перехватывает исключения из RestTemplate.
PriceScanner	Сканирует продукты, если цена меняется, то отправляет сообщение пользователю.
CustomUserDetails	Адаптер моего класса User для Spring Security.
CustomUserDetailsService	Сервис для класса CustomUserDetails.
JwtFilter	Фильтр для поступающих в программу запросов для авторизации.
JwtProvider	Класс для обработки токена.
ProductService	Сервис для товаров.

UserService	Сервис для работы с пользователем.
NotipriceTelegramBot	Телеграм бот. Обрабатывает запросы пользователей в чате.
NotipriceApplication	Точка входа в приложение.

ПРИЛОЖЕНИЕ 3

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ

Таблица П2 - Описание полей, методов и свойств класса ServiceConfig.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
jwtFilter	private open val	JwtFilter	get	Фильтр для поступающих в программу запросов для авторизации
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
restTemplate	public open	RestTemplate	builder: RestTemplate Builder, connectTimeout: Long, readTimeout: Long	Создание экземпляра класса RestTemplate.
configure	protected open	Unit	http: HttpSecurity?	Конфигурация безопасности программы.
corsConfigurationSource	public open	CorsConfigurationSource		Конфигурация для CORS Policy.

Таблица ПЗ - Описание полей, методов и свойств класса ProductController.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
productService	private open	ProductService	get	Сервис для товаров.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
addProduct	public open	ProductDto	product: ProductDto, username: String	Создание нового товара.
updateProduct	public open	Unit	id: Long, product: ProductDto	Изменение данных о товаре.
deleteProduct	public open	Unit	id: Long	Удаление товара.
getProducts	public open	List<ProductDto>	username: String	Получение товара по пользовательскому имени.
getProductById	public open	ProductDto	id: Long	Получение товара по идентификатору.
getProductXPathByUrl	public open	String	url: String	Получение xpath продукта по URL. В базе данных ищутся xpath от базового домена URL. Найденные xpath проверяются можно ли получить значение по этому xpath значение. Если таких несколько, то выбирается самый популярный.
getHtmlWithHighlightedElement	public open	String	url: String, xpath: String	Получает страницу по URL и выделяет элемент по xpath и возвращает страницу.

Таблица П4 - Описание полей, методов и свойств класса UserController.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
userService	private open	UserService	get	Сервис для работы с пользователем.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
updateUser	public open	Unit	id: Long, user: UserDto	Изменение данных о пользователе.
deleteUser	public open	Unit	id: Long	Удаление пользователя.
getUserByUsername	public open	UserDto	username: String	Получение данных о пользователе по пользовательскому имени.
getUserById	public open	UserDto	id: Long	Получение данных о пользователе по идентификатору.

Таблица П5 - Описание полей, методов и свойств класса ProductDao.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
jdbcTemplate	private open	JdbcTemplate	get	Обертка над JDBC
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
save	public open	Product	product: Product	Сохранение экземпляра класса Product в базе данных.
findByIdOrNull	public open	Product?	id: Long	Получение продукта по идентификатору.
delete	public open	Unit	productId: Long	Удаление продукта.
findAllUserProducts	public open	List<Product>	username: String	Получение товаров пользователя по пользовательскому имени.
findToCheck	public open	List<Product>	timeIntervalIn Seconds: Int, limit: Int	Получение товаров для сканирования. Возвращает товары, которые не проверялись определенный интервал времени в секундах timeInterval.
findXpathByUrl	public open	List<String>	baseUrl: String	Получение xpath-ов по URL.

Таблица П6 - Описание полей, методов и свойств класса SubscriptionDao.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
jdbcTemplate	private open	JdbcTemplate	get	Обертка над JDBC
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
save	public open	Subscription	subscription: Subscription	Добавление в базу данных экземпляра класса Subscription.
findByIdOrNull	public open	Subscription?	chatId: Long, productId: Long	Получение Subscription по идентификатору.
update	public open	Unit	product: Subscription	Изменение данных Subscription.
delete	public open	Unit	chatId: Long, productId: Long	Удаление Subscription.
findChatIdsByProductId	public open	List<Long>	productId: Long	Получение идентификаторов чата по идентификатору продуктов.

Таблица П7 - Описание полей, методов и свойств класса UserDao.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
jdbcTemplate	private open	JdbcTemplate	get	Обертка над JDBC
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
save	public open	User	user: User	Добавление в базу данных экземпляра класса User.
findByIdOrNull	public open	User?	id: Long	Получение User по идентификатору.
findByNameOrNull	public open	User?	name: String	Получение User по пользовательскому имени.
update	public open	Unit	user: User	Изменение данных User.
delete	public open	Unit	userId: Long	Удаление User.

Таблица П8 - Описание полей, методов и свойств дата класса ProductDto.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
id	public final	Long	get, set	Идентификатор товара.
name	public final	String	get, set	Название товара.
price	public final	Double	get, set	Цена товара в виде десятичного числа.
currency	public final	String	get, set	Валюта.
url	public final	String	get, set	URL на товар.
xpath	public final	String	get, set	Путь до цены на странице товара.
priceStr	public final	String	get, set	Значение по пути цены товара на странице.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение

Таблица П9 - Описание полей, методов и свойств класса UserDto.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
chatId	public final	Long	get	Идентификатор чата пользователя в Телеграме. Первичный ключ.
username	public final	String	get	Логин пользователя.
password	public final	String	get, set	Пароль пользователя.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение

Таблица П10 - Описание полей, методов и свойств класса Product.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
id	public final	Long	get, set	Идентификатор товара.
name	public final	String	get, set	Название товара.
price	public final	Double	get, set	Цена товара в виде десятичного числа.
currency	public final	String	get, set	Валюта.
url	public final	String	get, set	URL на товар.
xpath	public final	String	get, set	Путь до цены на странице товара.
priceStr	public final	String	get, set	Значение по пути цены товара на странице.
lastCheck	public final	Long	get, set	Timestamp времени последней проверки цены товара.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение

Таблица П11 - Описание полей, методов и свойств класса Subscription.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
chatId	public final	Long	get	Внешний ключ таблицы users.
productId	public final	Long	get	Внешний ключ таблицы products.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение

Таблица П12 - Описание полей, методов и свойств класса User.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
chatId	public final	Long	get	Внешний ключ таблицы users.
username	public final	String	get	Логин пользователя.
password	public final	String	get, set	Пароль пользователя.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение

Таблица П13 - Описание полей, методов и свойств класса ControllerExceptionHandler.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
handleIllegalArgumentExc eption	public open	Map<String, String>	e: IllegalArgume ntException	Перехватывает исключения типа IllegalArgumentException.
handleNoSuc hElementExc eption	public open	Map<String, String>	e: NoSuchEleme ntException	Перехватывает исключения типа NoSuchElementException.
handleResour ceAccessExce ption	public open	Map<String, String>	e: ResourceAcce ssException	Перехватывает исключения типа ResourceAccessException.
handleExcepti on	public open	Map<String, String>	e: Exception	Перехватывает исключения типа Exception.
errorResponse	private open	Map<String, String>	e: Exception	Создание ответа с информацией об ошибке.

Таблица П14 - Описание полей, методов и свойств класса

RestTemplateResponseErrorHandler.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
hasError	public open	Boolean	httpResponse: ClientHttpResponse	Является ли ответ ошибкой.
handleError	public open	Unit	httpResponse: ClientHttpResponse	Обработка ошибки.

Таблица П15 - Описание полей, методов и свойств класса PriceScanner.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
productDao	private open	ProductDao	get	DAO продукта для работы с базой данных.
subscriptionDao	private open	SubscriptionDao	get	DAO для класса Subscription для работы с базой данных.
restTemplate	private open	RestTemplate	get	Клиент для HTTP запросов.
timeIntervalInSeconds	private open	Int	get	Интервал времени в секундах. Показывает интервал проверки продуктов.
limit	private open	Int	get	Сколько товаров обрабатывать одновременно.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение

Таблица П16 - Описание полей, методов и свойств класса AuthController.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
userService	private open	UserService	get	Сервис для работы с пользователем.
jwtProvider	private open	JwtProvider	get	Класс для обработки токена.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
addUser	public open	UserDto	user: UserDto	Регистрация пользователя.
login	public open	String	user: UserDto	Проверяет пароль пользователя, если пароли совпадают, возвращает токен, если нет, то бросает исключение.

Таблица П17 - Описание полей, методов и свойств класса CustomUserDetails.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
login	private final	String	get	Логин пользователя.
password	private final	String	get	Пароль пользователя.
grantedAutho rities	private final	MutableColl ection<out GrantedAuth ority>	get	Роли пользователя.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение

Таблица П18 - Описание полей, методов и свойств класса CustomUserDetailsService.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
userService	private open	UserService	get	Сервис для работы с пользователем.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
loadUserByUsername	public open	CustomUserDetails	username: String?	олучение экземпляра класса CustomUserDetails по пользовательскому имени.

Таблица П19 - Описание полей, методов и свойств класса JwtFilter.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
jwtProvider	private open	JwtProvider	get	Класс для обработки токена.
customUserDetailsService	private open	CustomUserDetailsService	get	Сервис для класса CustomUserDetails.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
doFilter	public open	Unit	request: ServletRequest?, response: ServletResponse?, chain: FilterChain?	Фильтр для поступающих в программу запросов для авторизации.
getTokenFromRequest	Private fun	String?	request: HttpServletRequest	Получение токена из запроса.

Таблица П20 - Описание полей, методов и свойств класса JwtProvider.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
secret	private open	String	get	Секретный ключ для шифрования.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
generateToken	public open	String	login: String	Генерация токена.
validateToken	public open	Boolean	token: String	Валидация токена.
getLoginFromToken	public open	String	token: String	Получение пользовательского имени из токена.

Таблица П21 - Описание полей, методов и свойств класса ProductService.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
productDao	private open	ProductDao	get	DAO продукта для работы с базой данных.
subscriptionDao	private open	SubscriptionDao	get	DAO для класса Subscription для работы с базой данных.
userService	private open	UserService	get	Сервис для работы с пользователем.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
addProduct	public open	Product	product: Product, username: String	Добавление товара. Функция является транзакционной.
getProductById	public open	Product	id: Long	Получение товара по идентификатору.
getAllUserProducts	public open	List<Product>	List<Product>	Получение продуктов пользователя.
updateProduct	public open	Unit	product: Product	Изменение данных о товаре.
deleteProduct	public open	Unit	id: Long	Удаление продукта.
getHtmlWithHighlightedElement	public open	String	url: String, xpath: String	String
getProductXPathByUrl	public open	String	urlString: String	Получение xpath продукта по URL. В базе данных ищутся xpath от базового домена URL.Найденные xpath проверяются можно ли получить значение по этому xpath значение. Если таких несколько, то выбирается самый популярный.

Таблица П22 - Описание полей, методов и свойств класса

Поля				
Имя	Модификатор доступа	Тип	Назначение	
passwordEncoder	private open val	BCryptPasswordEncoder	Для хеширования пароля.	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
userDao	private open	UserDao	get	DAO пользователей для работы с базой данных.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
addUser	public open	User	user: User	Добавление пользователя.
getProductById	public open	User	id: Long	Получение пользователя по идентификатору.
getUserByUsername	public open	User	username: String	Получение пользователя по пользовательскому имени.
login	public open	User	user: User	Проверяет пароль пользователя, если пароли совпадают, возвращает пользователя, если нет, то бросает исключение.
updateUser	public open	Unit	user: User	Изменение данных о пользователе.
deleteProduct	public open	Unit	id: Long	Удаление пользователя.

Таблица ПЗ - Описание полей, методов и свойств класса NotipriceTelegramBot.

Поля				
Имя	Модификатор доступа	Тип	Назначение	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
botToken	private final	String	get	Токен Телеграм бота.
botUsername	private final	String	get	Пользовательское имя Телеграм бота.
notipriceUrl	private final	String	get	Ссылка на веб-приложение Notiprice. Используется для регистрации.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
getBotToken	public open	String		Геттер для botToken.
getBotUsername	public open	String		Геттер для botUsername.
onUpdateReceived	public open	Unit	update: Update	Обработка запросов пользователей.
handleSignUp	private final	Unit	chatId: String	Отправка ссылки для регистрации.

ПРИЛОЖЕНИЕ 4

ТЕРМИНОЛОГИЯ

Telegram бот – специальная программа, которая обрабатывает запросы пользователь через чат в Телеграмме.

Xpath (XML Path Language) – язык запросов к элементам XML-документа.

Контроллер – компонент из модели MVC, который отвечает за связь между Model и View.

Сервис – класс, который отвечает за бизнес логику. Нужен для разделения ответственности контроллера.

DAO (Data Access Object) – шаблон проектирования, абстракция для работы с базой данных.

DTO (Data Transfer Object) – шаблон проектирования, используется для обмена данными между системами.

JDBC (Java Database Connectivity) – стандартное взаимодействие Java/Kotlin приложений с различными СУБД.

[illegible]