Technical description

Ski online store

Author:

Shalnova Maria

Saint-Petersburg, 2019

Content.

1. Task.

To develop web-application that simulates the work of the online store information system. The application have to perform the required user`s cases.

User cases:

- For clients:
  - To view catalog with the possibility of filtering by parameters;
  - To view and edit profile;
  - To view and edit user addresses;
  - To view and edit user password;
  - Checkout;
  - To view order history;
- For managers:
  - To view clients orders;
  - To edit delivery status;
  - To view sales statistic (top 10 products, clients, proceeds per week, month);
  - To add new products;
  - To add and manage catalog categories.

Additionally, to develop co-application for advertisement generating from main application.

2. Project goals.

- The robust, useful and reliable system.
- Cohesive data model.
- User-friendly interface.
- Separate access to different system`s part.

3. Application description.

Web-application has two type of user: clients and managers.

Clients can view catalog, products, add products to cart and make an order. Clients also can change personal information, add and remove addresses, view order history.

Managers can add, delete products, manage categories. Managers also can looking for list of clients orders and edit their delivery status. Managers have clients options too.

There is an authentication mechanism in system that control access to portal. Each user in application has access level that display what information he could get and what couldn`t.

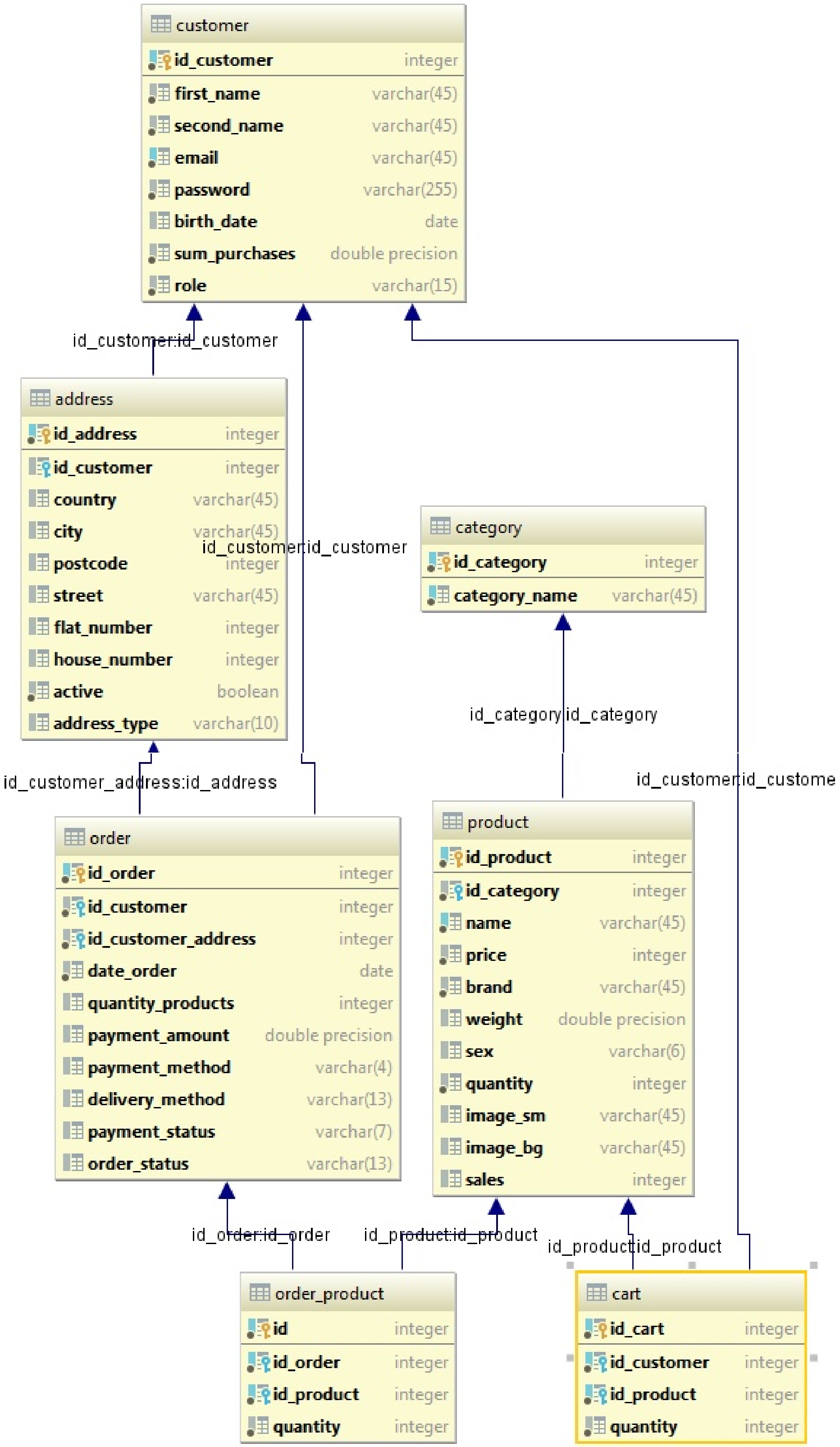
Data of users and their options store in reliable database.

4. Used Technologies.

- Instruments:
  - IDE - IntelliJ IDEA 2018 3.5
  - Maven 3.3.9

- Technologies:
  - Apache ActiveMQ Artemis 5.15.3
  - Ajax
  - Bootstrap 4.3.1
  - DB - PostgreSQL 8.4
  - EJB 3
  - Gson 2.8.5
  - Java 8
  - Javascript
  - JPA 2.0
  - JSF 2.3.2
  - JSP 2.1
  - Junit 4.12
  - Log4j 1.2.17
  - Mockito 2.10.0
  - Omnifaces 2.6.1
  - Bootsfaces 1.1.0
  - REST
  - Spring 5.1.5
  - Spring Security 5.1.4
  - Tomcat 8.5.38
  - WildFly 16.0.0

5. Database model.



customer	
id_customer	integer
first_name	varchar(45)
second_name	varchar(45)
email	varchar(45)
password	varchar(255)
birth_date	date
sum_purchases	double precision
role	varchar(15)

address	
id_address	integer
id_customer	integer
country	varchar(45)
city	varchar(45)
postcode	integer
street	varchar(45)
flat_number	integer
house_number	integer
active	boolean
address_type	varchar(10)

category	
id_category	integer
category_name	varchar(45)

order	
id_order	integer
id_customer	integer
id_customer_address	integer
date_order	date
quantity_products	integer
payment_amount	double precision
payment_method	varchar(4)
delivery_method	varchar(13)
payment_status	varchar(7)
order_status	varchar(13)

product	
id_product	integer
id_category	integer
name	varchar(45)
price	integer
brand	varchar(45)
weight	double precision
sex	varchar(6)
quantity	integer
image_sm	varchar(45)
image_bg	varchar(45)
sales	integer

order_product	
id	integer
id_order	integer
id_product	integer
quantity	integer

cart	
id_cart	integer
id_customer	integer
id_product	integer
quantity	integer

Order consists on "carts" which has data about one product, count of this product , customer who put product in cart.

6. System infrastructure.

- Front-end (browser presentation level):

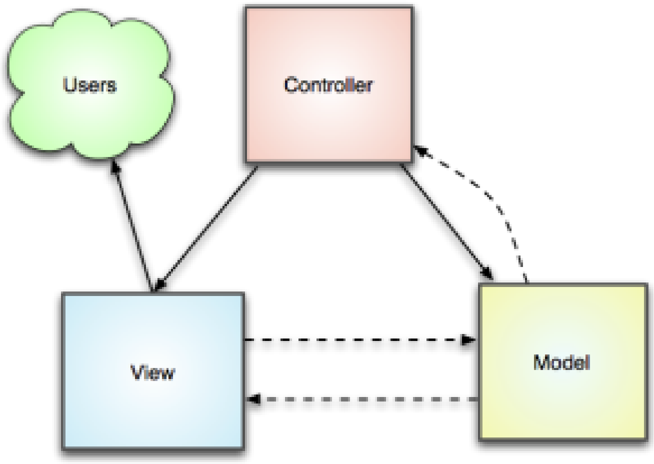
    1) Web-page structure - HTML
    2) Page-design - CSS
    3) Dynamic content - JavaScript, Ajax.

- Back-end (server based level):
    1) Application server - WildFly
    2) Database - PostgreSQL
    3) Server logic - Spring Framework
    4) ORM - Hibernate provider

- Client advertisement application:
    1) Web-pages - JSF
    2) JMS - Apache ActiveMQ Artemis imbedded in WildFly
    3) Application server - WildFly
    4) Server logic - EJB
    5) WS - REST

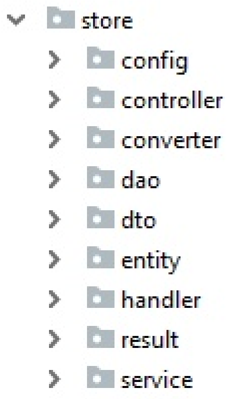
7. System architecture.

Architecture of server-based part presented by MVC - design pattern.



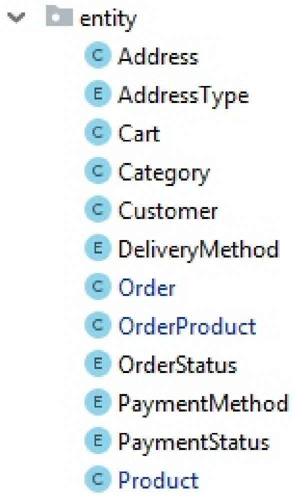
Class structure.

According MVC-pattern application has next structure:



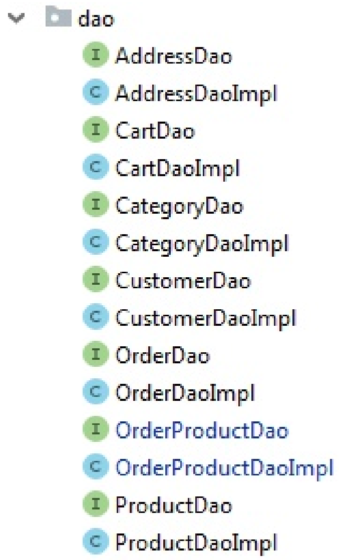
- store
  - config
  - controller
  - converter
  - dao
  - dto
  - entity
  - handler
  - result
  - service

Model level:



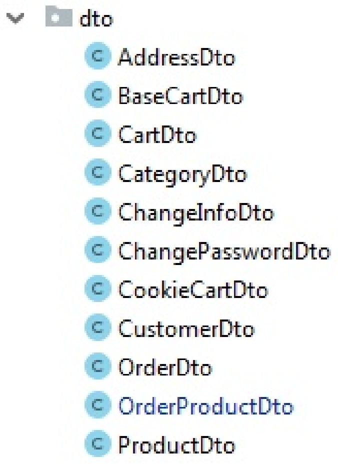
```
entity
    C Address
    E AddressType
    C Cart
    C Category
    C Customer
    E DeliveryMethod
    C Order
    C OrderProduct
    E OrderStatus
    E PaymentMethod
    E PaymentStatus
    C Product
```

Model-service level:



```
dao
    I AddressDao
    C AddressDaoImpl
    I CartDao
    C CartDaoImpl
    I CategoryDao
    C CategoryDaoImpl
    I CustomerDao
    C CustomerDaoImpl
    I OrderDao
    C OrderDaoImpl
    I OrderProductDao
    C OrderProductDaoImpl
    I ProductDao
    C ProductDaoImpl
```

Data transport objects:

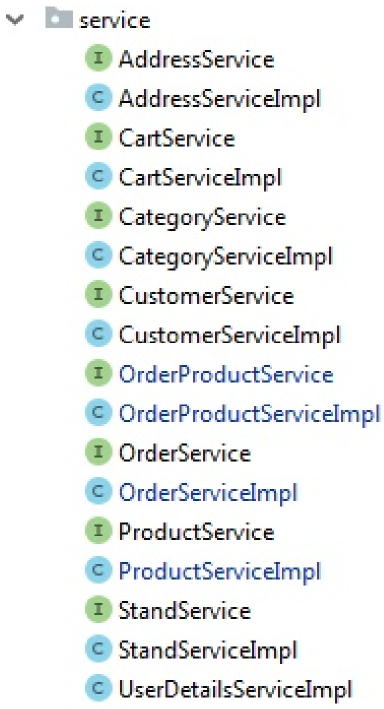


```
dto
    C AddressDto
    C BaseCartDto
    C CartDto
    C CategoryDto
    C ChangeInfoDto
    C ChangePasswordDto
    C CookieCartDto
    C CustomerDto
    C OrderDto
    C OrderProductDto
    C ProductDto
```

Each DAO class has queries to database. Developer has 3 ways to create query:

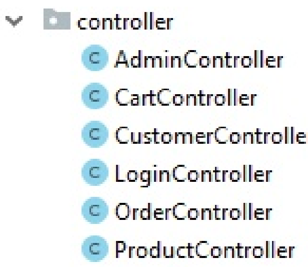
1. Native query
2. Hibernate query language
3. Criteria API

Service level:



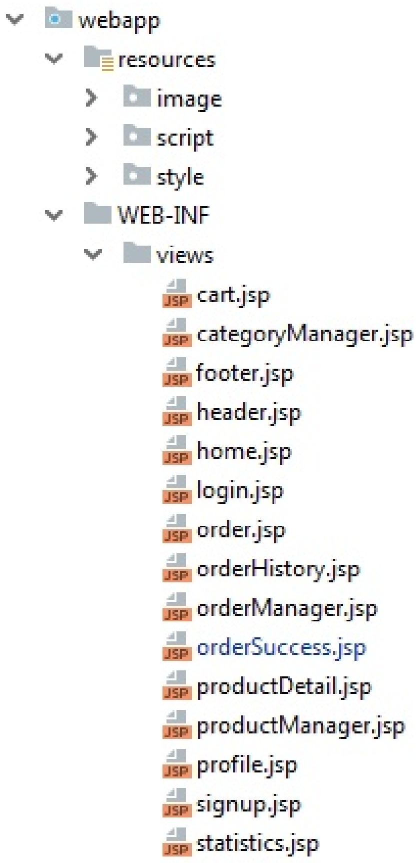
```
service
    I AddressService
    C AddressServiceImpl
    I CartService
    C CartServiceImpl
    I CategoryService
    C CategoryServiceImpl
    I CustomerService
    C CustomerServiceImpl
    I OrderProductService
    C OrderProductServiceImpl
    I OrderService
    C OrderServiceImpl
    I ProductService
    C ProductServiceImpl
    I StandService
    C StandServiceImpl
    C UserDetailsServiceImpl
```

View-service level:



```
controller
    C AdminController
    C CartController
    C CustomerController
    C LoginController
    C OrderController
    C ProductController
```

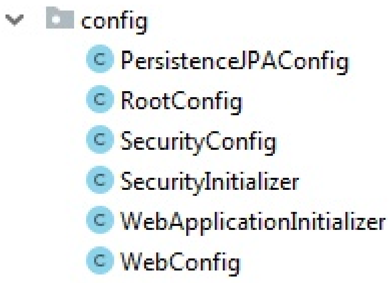
View level:



```
webapp
  resources
    image
    script
    style
  WEB-INF
    views
      cart.jsp
      categoryManager.jsp
      footer.jsp
      header.jsp
      home.jsp
      login.jsp
      order.jsp
      orderHistory.jsp
      orderManager.jsp
      orderSuccess.jsp
      productDetail.jsp
      productManager.jsp
      profile.jsp
      signup.jsp
      statistics.jsp
```

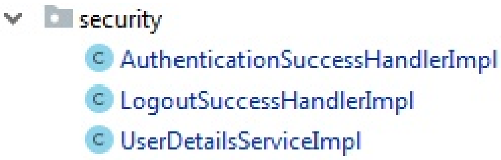
Directory *style* contains cssfiles.
Directory scripts contains javascripts files.
Directory image contains static images.

Configuration:



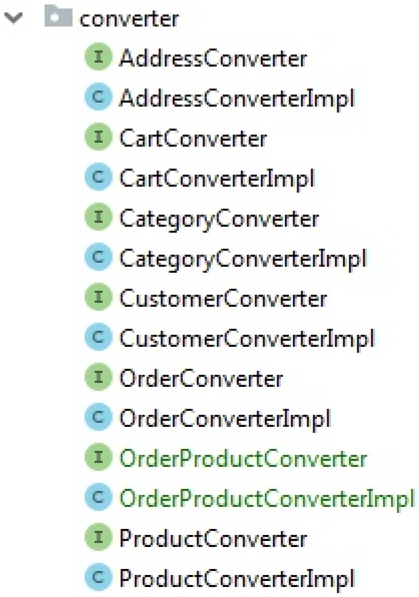
```
config
  PersistenceJPAConfig
  RootConfig
  SecurityConfig
  SecurityInitializer
  WebApplicationInitializer
  WebConfig
```

Security mechanism:

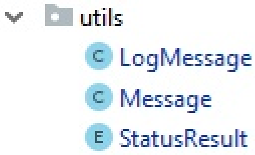


```
security
    AuthenticationSuccessHandlerImpl
    LogoutSuccessHandlerImpl
    UserDetailsServiceImpl
```

Support utilities:



```
converter
    AddressConverter
    AddressConverterImpl
    CartConverter
    CartConverterImpl
    CategoryConverter
    CategoryConverterImpl
    CustomerConverter
    CustomerConverterImpl
    OrderConverter
    OrderConverterImpl
    OrderProductConverter
    OrderProductConverterImpl
    ProductConverter
    ProductConverterImpl
```

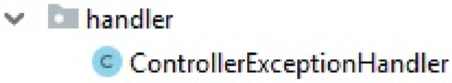


```
utils
    LogMessage
    Message
    StatusResult
```

converter contains classes to convert entity to dto objects, and dto to entity.

utils contains static content for messages, status codes.

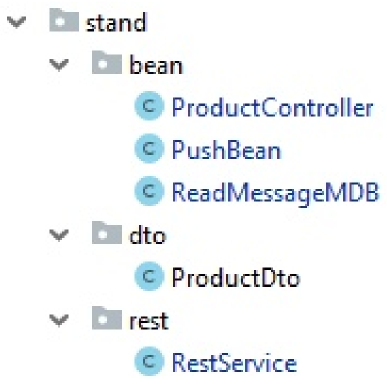
Exceptions handler:



```
handler
    ControllerExceptionHandler
```

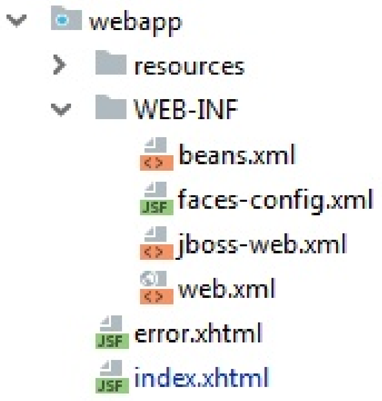
Rest client application integrated in main application interface and build on EJB and JSF technologies.

Business logic:



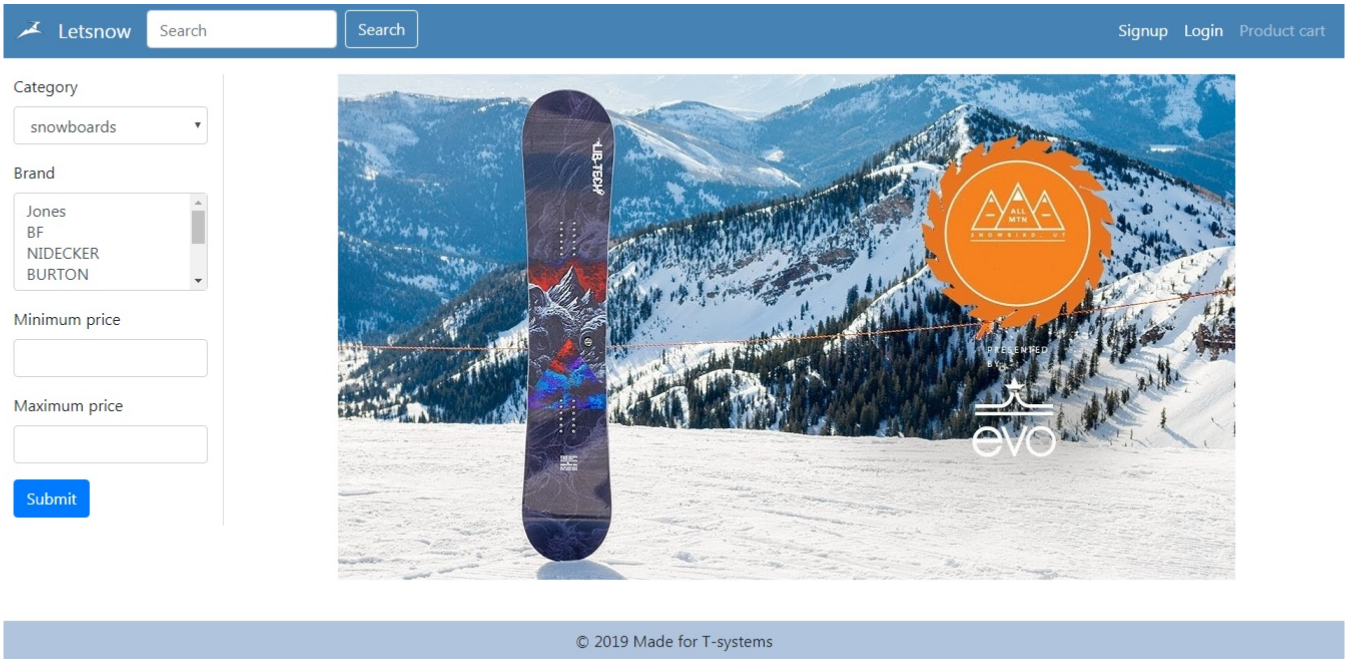
```
stand
  bean
    ProductController
    PushBean
    ReadMessageMDB
  dto
    ProductDto
  rest
    RestService
```

View and xml configurations:



```
webapp
  resources
  WEB-INF
    beans.xml
    faces-config.xml
    jboss-web.xml
    web.xml
  error.xhtml
  index.xhtml
```

8. UI.

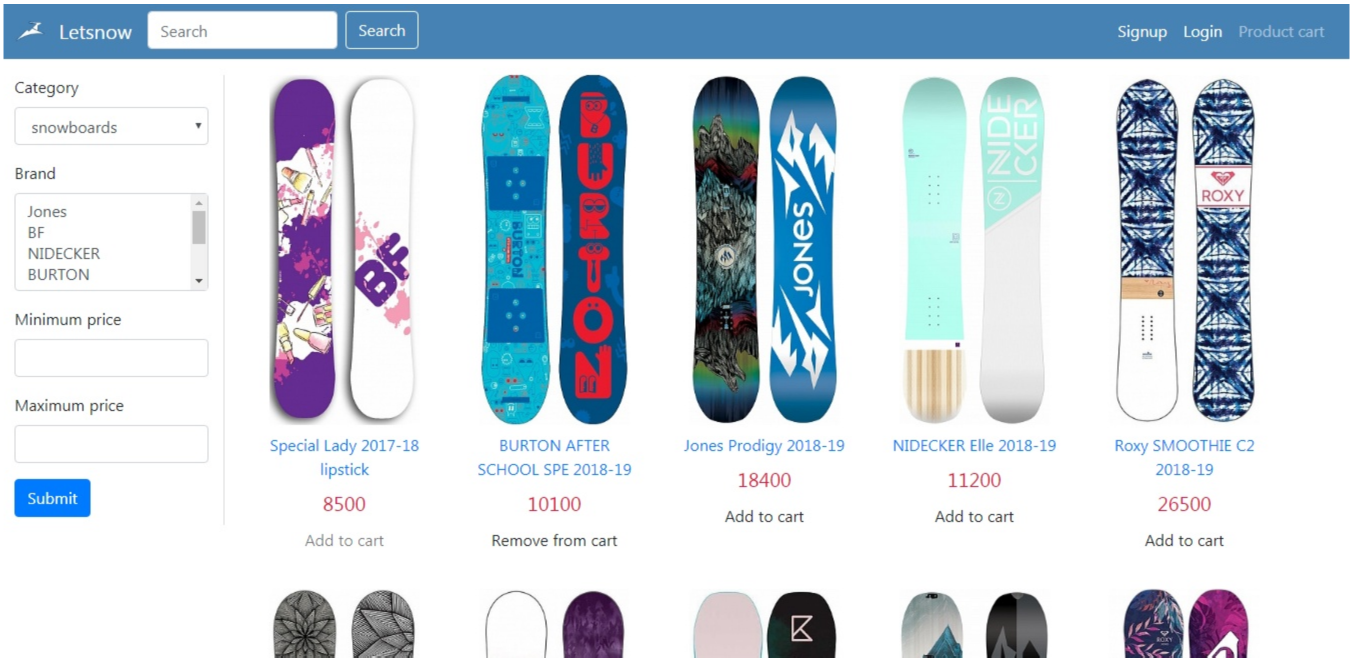


Category: snowboards

Brand: Jones, BF, NIDECKER, BURTON

© 2019 Made for T-systems

Picture 1. Home page

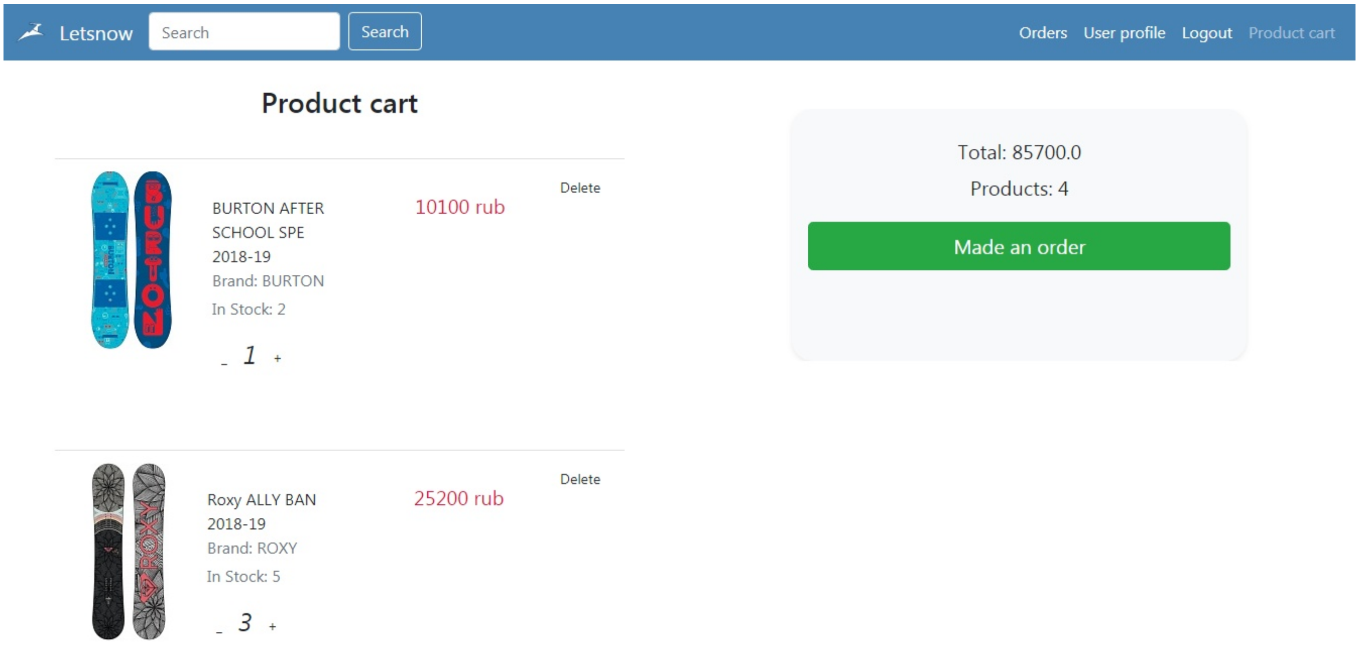


Category: snowboards

Brand: Jones, BF, NIDECKER, BURTON

Special Lady 2017-18 lipstick	8500	Add to cart
BURTON AFTER SCHOOL SPE 2018-19	10100	Remove from cart
Jones Prodigy 2018-19	18400	Add to cart
NIDECKER Elle 2018-19	11200	Add to cart
Roxy SMOOTHIE C2 2018-19	26500	Add to cart

Picture 2. Catalog page



Product cart

BURTON AFTER SCHOOL SPE 2018-19
Brand: BURTON
In Stock: 2
_ 1 +

10100 rub

Delete

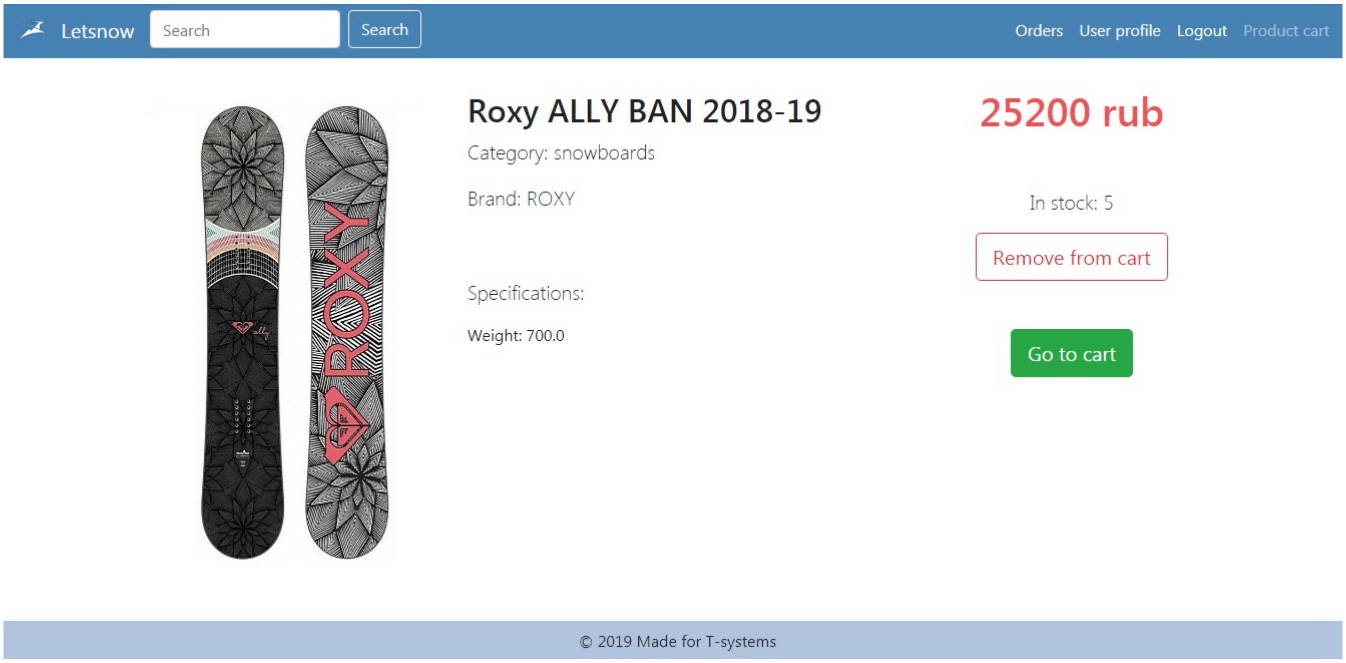
Roxy ALLY BAN 2018-19
Brand: ROXY
In Stock: 5
_ 3 +

25200 rub

Delete

Total: 85700.0
Products: 4

Picture 3. Product cart page



Roxy ALLY BAN 2018-19

Category: snowboards

Brand: ROXY

Specifications:

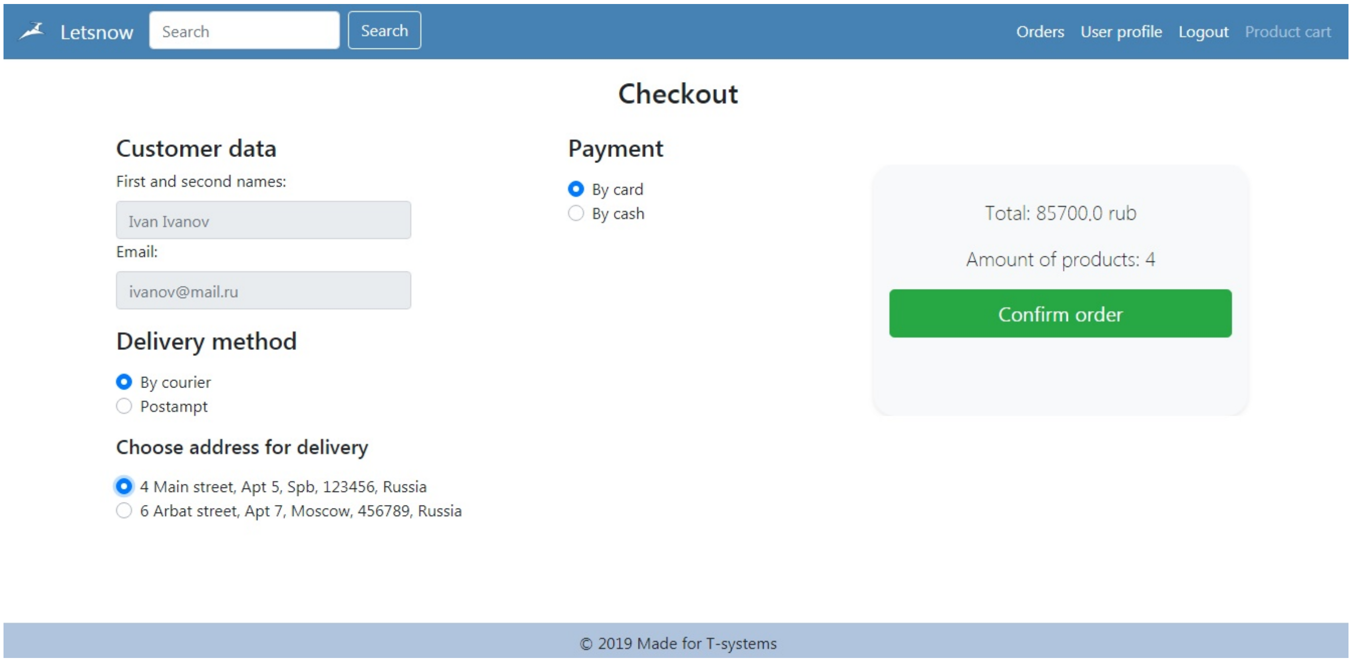
Weight: 700.0

25200 rub

In stock: 5

© 2019 Made for T-systems

Picture 4. Checkout page



Checkout

Customer data

First and second names: Ivan Ivanov

Email: ivanov@mail.ru

Delivery method

● By courier
○ Postampt

Choose address for delivery

● 4 Main street, Apt 5, Spb, 123456, Russia
○ 6 Arbat street, Apt 7, Moscow, 456789, Russia

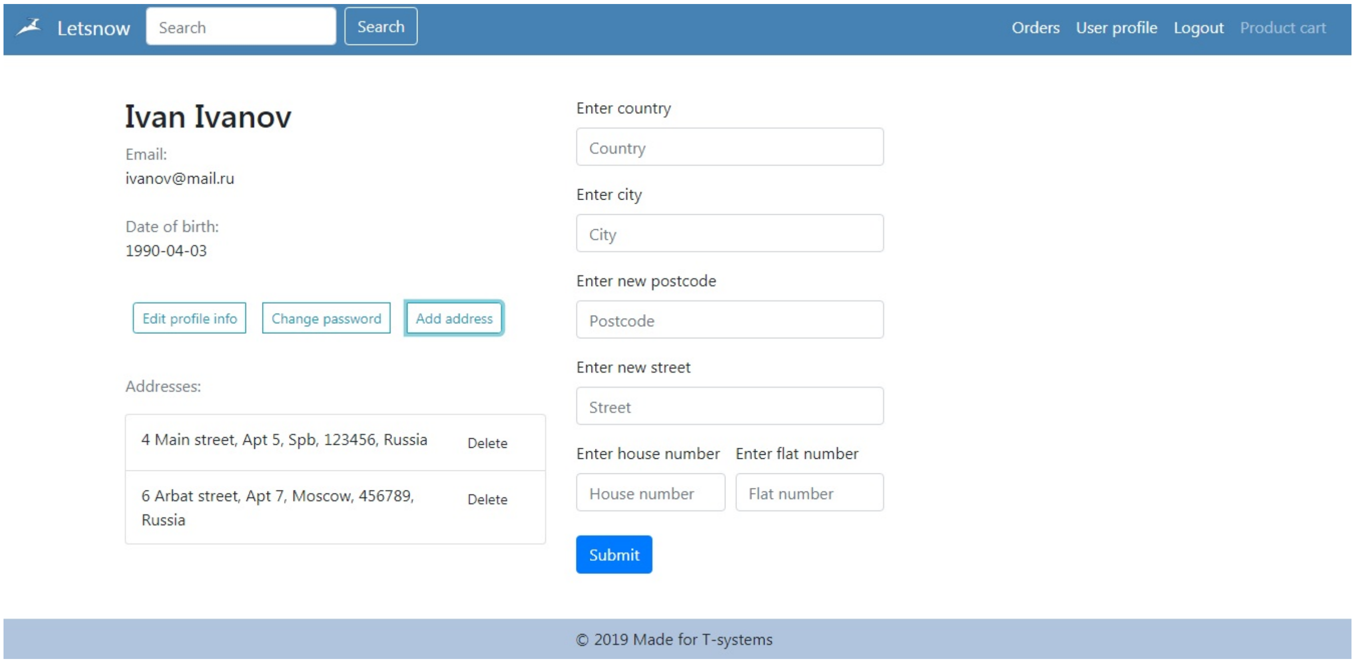
Payment

● By card
○ By cash

Total: 85700.0 rub

Amount of products: 4

Picture 5. Checkout page



Ivan Ivanov

Email: ivanov@mail.ru

Date of birth: 1990-04-03

Addresses:

4 Main street, Apt 5, Spb, 123456, Russia	Delete
6 Arbat street, Apt 7, Moscow, 456789, Russia	Delete

Enter country: Country

Enter city: City

Enter new postcode: Postcode

Enter new street: Street

Enter house number: House number

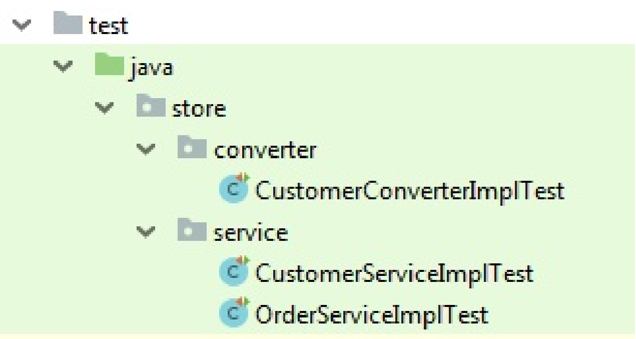
Enter flat number: Flat number

Picture 6. Customers profile page

9. Code quality.

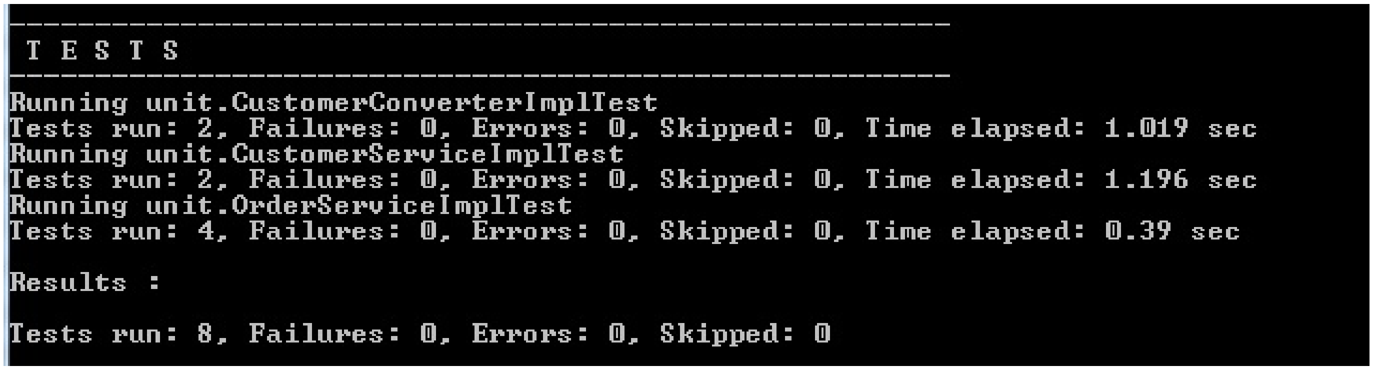
Tests.

Test structure:



```
test
  java
    store
      converter
        CustomerConverterImplTest
      service
        CustomerServiceImplTest
        OrderServiceImplTest
```

JUnit tests:



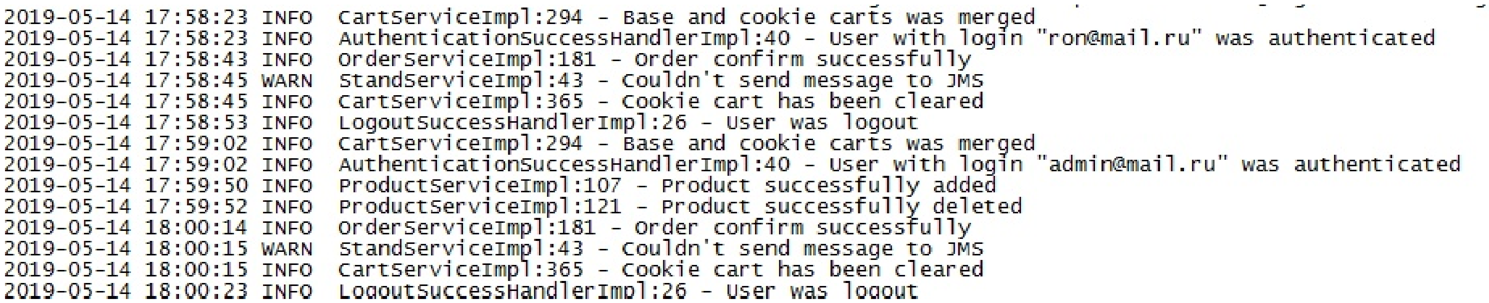
```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running unit.CustomerConverterImplTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.019 sec
Running unit.CustomerServiceImplTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.196 sec
Running unit.OrderServiceImplTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.39 sec

Results :

Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
```

Logging.

C:/tmp/log_file.txt



```
2019-05-14 17:58:23 INFO  CartServiceImpl:294 - Base and cookie carts was merged
2019-05-14 17:58:23 INFO  AuthenticationSuccessHandlerImpl:40 - User with login "ron@mail.ru" was authenticated
2019-05-14 17:58:43 INFO  OrderServiceImpl:181 - Order confirm successfully
2019-05-14 17:58:45 WARN  StandServiceImpl:43 - Couldn't send message to JMS
2019-05-14 17:58:45 INFO  CartServiceImpl:365 - Cookie cart has been cleared
2019-05-14 17:58:53 INFO  LogoutSuccessHandlerImpl:26 - User was logout
2019-05-14 17:59:02 INFO  CartServiceImpl:294 - Base and cookie carts was merged
2019-05-14 17:59:02 INFO  AuthenticationSuccessHandlerImpl:40 - User with login "admin@mail.ru" was authenticated
2019-05-14 17:59:50 INFO  ProductServiceImpl:107 - Product successfully added
2019-05-14 17:59:52 INFO  ProductServiceImpl:121 - Product successfully deleted
2019-05-14 18:00:14 INFO  OrderServiceImpl:181 - Order confirm successfully
2019-05-14 18:00:15 WARN  StandServiceImpl:43 - Couldn't send message to JMS
2019-05-14 18:00:15 INFO  CartServiceImpl:365 - Cookie cart has been cleared
2019-05-14 18:00:23 INFO  LogoutSuccessHandlerImpl:26 - User was logout
```

12. Future improvement.

1. Adding new functionality (payment system, etc.).
2. Use docker for runnig application
3. Refactoring and optimization code.

4. Adding integrations tests