

Final Task

Online store information system.

There are the following types of entities:

- Order
 - o Client
 - o Client address
 - o Payment method (Cash \ By card (optional, make a trivial emulation of the payment system))
 - o Delivery method
 - o Goods
 - o Payment status (pending/paid)
 - o Order status (pending payment/pending shipment/shipped/delivered)
- Product
 - o Title
 - o Price
 - o Category
 - o Parameters (brand\color\weight...)
 - o Weight
 - o Volume
 - o Quantity in stock
- Client
 - o Name
 - o Surname
 - o Date of birth
 - o Email address
 - o Password for personal account
- Client's address
 - o Country
 - o City
 - o Postal code
 - o Street
 - o Home
 - o Apartment

The application must provide the following functionality:

- For company clients
 - o Catalog browsing with the ability to filter by parameters
 - o Viewing and editing a profile
 - Information
 - Addresses
 - Password
 - o Orders
 - Placing an order
 - View order history
 - (Optional) Reorder
- For company employees
 - o Orders
 - View
 - Changing the order status (shipped\delivered\paid)
 - o Sales statistics (top 10 products, customers, monthly/weekly revenue)
 - o Goods
 - Adding
 - Create and manage catalog categories
 - (Optional) import from file

When making a purchase, each page must display a shopping cart displaying the user's selected items prior to checkout. The cart is displayed for both the guest and the logged in user. Moreover, after authorization, the contents of the basket are not lost. After closing the page, the basket is not cleared.

Technical requirements

As a result, it is required to obtain a multi-user client-server application with a network connection.

All data is stored on the server side. Each client can upload some data, after each change operation the data must be synchronized with the server.

The client must have a graphical interface.

The application must handle hardware and software errors.

Criteria for successful completion of the task

1. Functionality works (UI required)
2. Maven-based project divided into modules (build with one command, deploy with one command)
3. Domain interfaces are described
4. Connected MySQL database
5. Entities of the subject area are created; mapping to tables in the database
6. Working with entities through DAO
7. Application deployed on AS
8. Implemented exception handling
9. Logging enabled
10. Availability of technical solution description
11. Availability of unit tests for business logic

Technologies for project implementation

Within the framework of the school, students are not limited in the choice of technologies for project implementation within the technology stack used in the company in various projects (lectures cover a small part of them). The list of technologies to choose from is given below. However, the project must be written primarily in java (minor parts, separate modules, or additional functionality can be written in other languages, such as Kotlin or Python). After discussion with the mentor and in agreement with him, the student can choose any combination of technologies and frameworks, but you should know that the mentor may be unfamiliar with some of them.

List of technologies used in some of the company's projects:

Spring Core, Spring Data, AOP, Spring Boot, Spring Security, MVC, Spring Cloud, Microservices, Data processing (Spark, Apache Flume), CDI/EJB, Apache CXF, RedHat Fuse, Oracle eCommerce (ATG, Endeca...), JBoss, Apache Tomcat, Embedded Tomcat H2 (in-memory), Websphere Application Server 9.0, React, Angular/NPM/Webpack, typescript, JSF/ExtJs, JSP/Servlets, JavaScript / TypeScript / HTML / Sass / CSS/LESS/jQuery/GraphQL, SOAP/REST, JDBC, XML/XSLT/XSD, Bash scripting, Oracle Database, Gradle, Git, Maven, Subversion, Jenkins, Grafana, GitlabCI, Sonar, UNIX shell, Bootstrap, Formbased authorisation/JWT, Drools, ARS, DOM, PL SQL, Oracle AQ, IBM MQ, JMS, OracleDB, Apache Camel, Docker, Microsoft T-Sql, WebDriver (Java), Oracle ATG, python, mapstruct, Lombok, Groovy, openshift, EC2, S3, Mongo, Casandra, hazelcast, Junit, testing, Selenium, Mockito, Selenid, Prometheus, Kubernetes, Helm, Kibana, AWS, Apache Camel, Go, K8s, ETCD, PostgreSQL, Wicket, blockchain, Web Flux, JPA/Hibernate

It will be a plus to use the following technologies: Selenium, Sonar, Angular / React, Docker, Microservices, using an available cloud to deploy an application and/or the presence of "killer features".