UDC 004.41

DEVELOPMENT OF ONLINE SERVICE FOR TEACHING PROGRAMMING ON JAVA

PAVEL STANKEVICH, YURI KRAVCHENKO
Polotsk State University, Belarus

The article deals with the step-by-step development of a web application with MVC architecture using the level of business logic (services), as well as POJO. This web application was implemented using a number of frameworks, including Spring Framework, Spring Security and MyBatis.

Introduction. Recently, online services have been gaining increasing popularity. They have a number of advantages, unlike traditional teaching methods. Online services can provide materials in the form convenient for perception: tables, graphics, animation, store a large amount of information of different kinds in one place and work with this information easily. They also provide an opportunity for transition to advanced teaching methods, when students study independently.

There is a small range of free online learning systems that are needed as a training apparatus at university. To such systems, including, belong systems for teaching programming languages. When analyzing the most popular languages and similar systems, the Java language was chosen, since it takes the second place in the ranking of the most popular programming languages after the query language SQL.

Main part. James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. Java was originally designed for interactive television, but it was too advanced for the digital cable television industry of the time. The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project got the name Green and was finally renamed Java, from Java coffee. Gosling designed Java with a C/C++style syntax that system and application programmers would find familiar.

PostgreSQL 9.6 was chosen as the database. This DBMS is a free software with an open source, and it is a very powerful system. There is a fairly large community where you can easily find answers to your questions. Despite the huge number of built-in functions, there are a lot of add-ons which allow you to manage the data [1].

List of the main frameworks used:

- Spring Framework. This is the central element of Spring is its inversion of control (IoC) container, which provides a consistent means of configuring and managing Java objects using reflection. The container is responsible for managing object lifecycles of specific objects: creating these objects, calling their initialization methods, and configuring these objects by wiring them together [2].

- Spring MVC. The Spring Framework features its own model-view-controller (MVC) web application framework, which wasn't originally planned. The Spring developers decided to write their own Web framework as a reaction to what they perceived as the poor design of the (then) popular Jakarta Struts Web framework, as well as deficiencies in other available frameworks. In particular, they felt there was insufficient separation between the presentation and request handling layers, and between the request handling layer and the model. Like Struts, Spring MVC is a request-based framework. The framework defines strategy interfaces for all of the responsibilities that must be handled by a modern request-based framework. The goal of each interface is to be simple and clear so that it's easy for Spring MVC users to write their own implementations, if they choose so. MVC paves the way for cleaner front-end code. All interfaces are tightly coupled to the Servlet API. This tight coupling to the Servlet API is seen by some as a failure on the part of the Spring developers to offer a high-level abstraction for Web-based applications [citation needed]. However, this coupling makes sure that the features of the Servlet API remain available to developers while offering a high abstraction framework to simplify work with the said API [2].

- Spring Security. This is a Java/Java EE framework that provides authentication, authorization and other security features for enterprise applications. The project was started in late 2003 as 'Acegi Security' by Ben Alex, with it being publicly released under the Apache License in March 2004. Subsequently, Acegi was incorporated into the Spring portfolio as Spring Security, an official Spring sub-project. The first public release under the new name was Spring Security 2.0.0 in April 2008, with commercial support and training available from SpringSource [2].

- Commons DBCP. This framework is responsible for creating a connection pool to a database in which each connection can consistently serve multiple clients.

- MyBatis. This is a Java persistence framework that couples objects with stored procedures or SQL statements using an XML descriptor or annotations [3].

To develop the client part, HTML was selected and the Bootstrap framework. It includes a set of tools for creating websites and web applications, HTML and CSS design templates for typography, web forms, buttons, labels, navigation blocks and other web interface components, including JavaScript extensions.

When designing a web-application, the architecture presented in Figure 1 was taken as the basis.

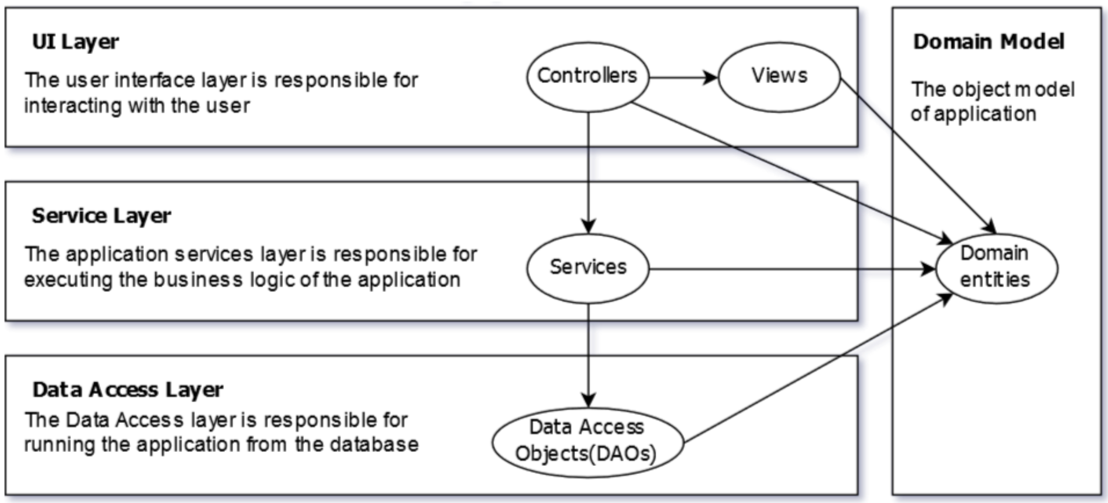
UI Layer
The user interface layer is responsible for interacting with the user
Controllers
Views
Service Layer
The application services layer is responsible for executing the business logic of the application
Services
Data Access Layer
The Data Access layer is responsible for running the application from the database
Data Access Objects(DAOs)
Domain Model
The object model of application
Domain entities

Fig. 1. The architecture of the developed web application

At the initial stage, the database schema was designed (Figure 2). From the figure it is possible to understand what functionality the developed service has.

After all the tables were defined, it was necessary to create the entity classes in the web application for convenient data manipulation. These classes are very similar to the tables in the database, but instead of storing the identifier for related data from another class, objects are stored. That is, if the product ID contains a record identifier from the product_types table, then the ProductType object is stored in the Product class object. In the case where many-to-many relationships are used, the entity list stores a list of objects.

The next step was the creation of DAO interfaces, as well as their implementation to retrieve and modify data stored in the database. Data Access Object is an object which provides an abstract interface to any type of database or storage mechanism. Certain features are provided regardless of what storage mechanism is used.

DAO was implemented with the help of MyBatis, for this purpose there were created classes that extend SqlSessionDaoSupport and implement their own DAO interface, as well as XML files where queries to the database were written. The implementation of methods consists mainly of getting the SQL session and calling the method described in the XML file.

When the DAO was ready and tested successfully, the implementation of the service layer (business logic) was started. As in DAO, first, a set of interfaces was created in which it was clearly defined what methods should be used. In the interface implementation classes, the fields are the DAO interfaces for which methods for inserting data (setters) were created. The methods used these fields to retrieve data from the database. As an example, you can consider the implementation of the process of saving a user in the system. Suppose we have a saveUser method in the service, it is necessary to save the user, and it does not matter if it exists at the moment, for this, the user's ID is checked in the implementation. If it is null, then it is necessary to call the create method from DAO, which will create a user, but in case the user already has an identifier, then the update method will be called, which will update the changed user data.

The next step was to create jsp pages, implement a set of custom tags, and also create controllers for the corresponding pages, as well as validators (on pages where data was stored in the database).

A custom tag is a tag that describes its implementation of something in HTML. For example, you can create an input tag, but where it will be specified additionally in which it will be wrapped (div with a certain set of classes), and will also have a label for this element (label). As parameters to such a custom tag, you can specify the text to the element's signature. These tags are very useful in case you are working with some UI framework, as usually simple designs are complicated by additional tags.

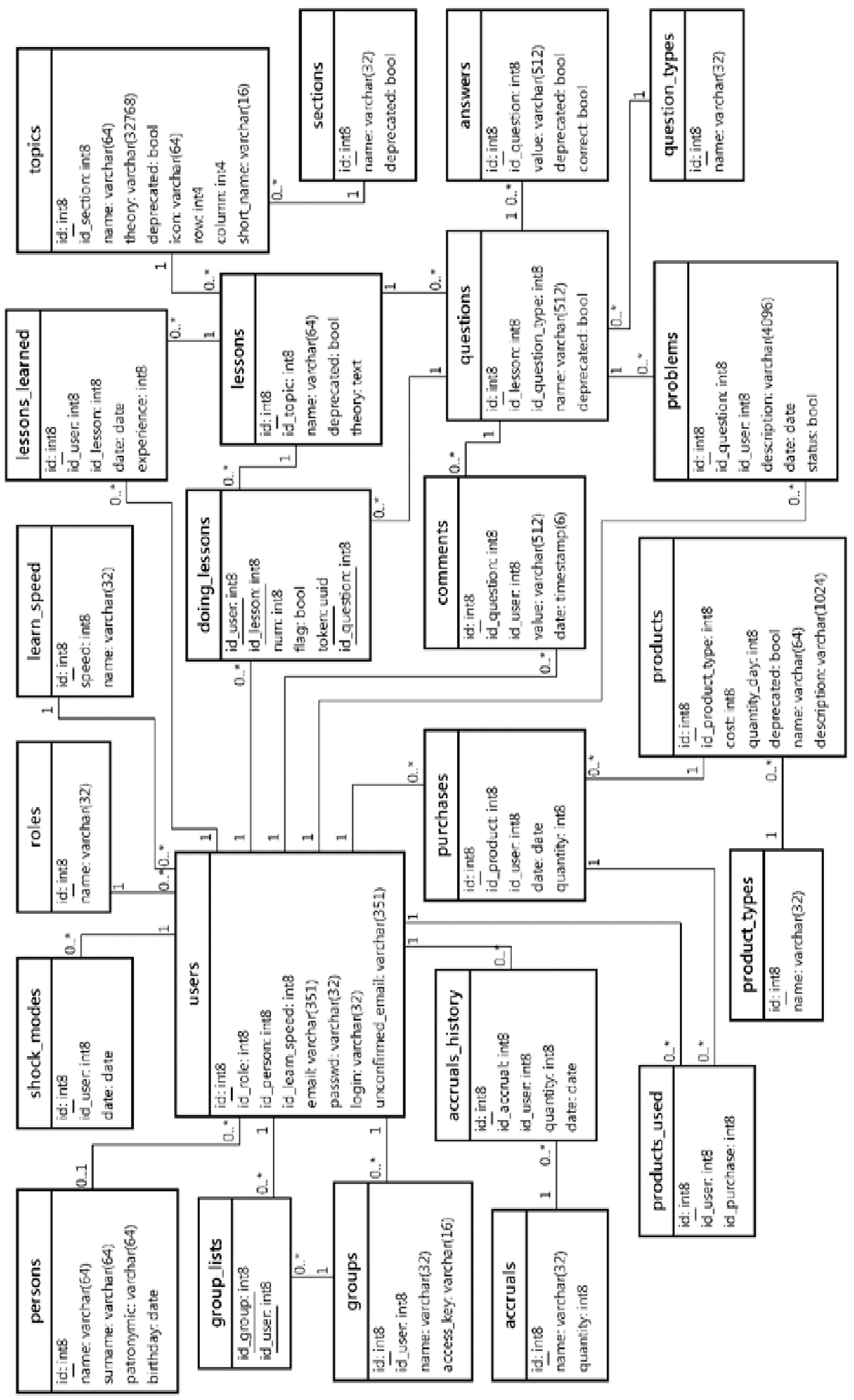

Fig. 2. Database schema of the developed web application

JSP (JavaServer Pages) is a technology that allows you to create content that has both static and dynamic components. The JSP page contains two types of text: static source data, which can be formatted in one of the text formats HTML, SVG, WML, or XML, and JSP elements that construct dynamic content. In addition, JSP-tag libraries, as well as EL (Expression Language), can be used to embed Java code in the static content of JSP pages.

Page controllers are implemented as follows. In the case when a stranger receives any data from the user, the validator is called, which checks the transmitted data. If all the data is correct, then the saving method is called, and if not, the user will receive an error of filling in the data. On pages that do not have the ability to update the data, depending on the situation, the data is transferred from the services to the JSP page.

Conclusion. Having considered the steps of creating this web application and architecture, it can be concluded that all were implemented in dependency order, as each layer of the web-application uses some other layer. It can also be noted that the author designed and implemented the service for interesting language learning of Java programming. Additional functionality is currently being developed.

REFERENCES

1. PostgreSQL – Wikipedia [Electronic resource]. – Mode of access: https://en.wikipedia.org/wiki/PostgreSQL. – Date of access: 14.12.2017.
2. Spring Framework – Wikipedia [Electronic resource]. – Mode of access: https://en.wikipedia.org/wiki/Spring_Framework. – Date of access: 14.12.2017.
3. MyBatis - Wikipedia [Electronic resource]. – Mode of access: https://en.wikipedia.org/wiki/MyBatis. – Date of access: 15.12.2017.