

Since in given case there is a specific enterprise project, the export subsystem includes a replicator that receives data from the queue and sends data to a specific repository. The JUnit framework, Testcontainers Java library and Spring JmsTemplate were chosen for the integration testing.

Spring JmsTemplate allows to emulate messages coming from the queue, which are subsequently processed by the business logic of the replication service, and then the data is sent to the Allegrograph repository located in the Docker container (Figure 1). This container starts working before the first test starts and is deleted after all tests are completed. Testcontainers library allows to manage lifecycle of Allegrograph container, set up version, image name, registry, ports etc. After the data has been saved in the Allegrograph, it must be subtracted using a SPARQL query and compared with the original data. In order to reduce reliance on serialization differences, Jena model comparison was used instead of string values.

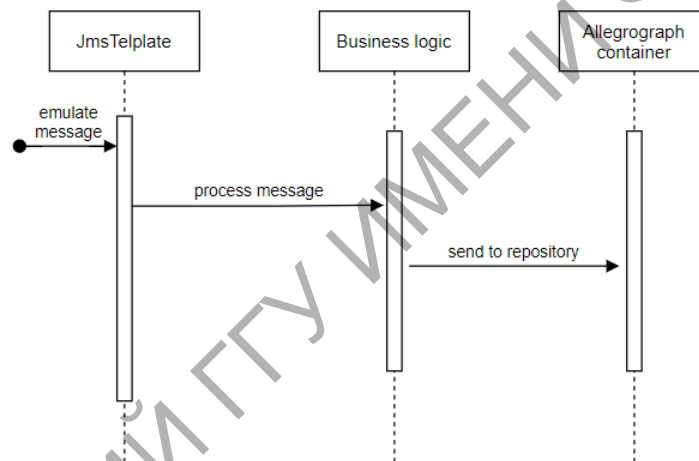


Figure 1 – Interaction scheme

Stepan Dubrovski

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

FUNCTIONAL OPPORTUNITIES OF A MOBILE CLIENT FOR A DISTRIBUTION NETWORK

Nowadays, the way of gaining access to a certain product or service has changed a lot, even over several years. Today it is impossible to conduct a business focused on the distribution of goods or the provision of services without using modern approaches to sales and consumption. Currently, retail

stores, manufacturers of goods and services are experiencing serious problems with the expansion of their sales network and consumers inflow. This happens due to several factors. The most significant is the transition from consumption directly when the consumer physically appears in the store to consumption when a product or service is purchased remotely, with the opportunity to get acquainted with the assortment without looking up from the handy gadget and without leaving home. This pattern of consumption has gained particular relevance due to recent restrictions on movement due to the epidemiological situation. In addition to this, people gained an understanding of the convenience of remote shopping and delivery at the time and place they need without having to move around the city on their own in search of the necessary product or service. In this regard, the business is forced to rebuild to the modern realities of the market and develop its own distribution network of goods or acquire such an opportunity on existing platforms.

There is a sufficient number of software products that give the opportunity of orders processing, product distribution and consumer notification in the process of some goods delivery. Some of them are overloaded and cannot be clear for an end consumer for each type of product. The main goal of this project was to create a better application, excluding an extra functionality overload.

This application is easy-to-use and provides a wide functional range. Doesn't require the installation of additional catalogues or market apps. Interfaces are created to give the end user a better perception and interaction. All this provides a good impression after interacting with the application and speeds up operations processing. It also gives a better chance that the consumer will come back to make purchases again.

The application development process was divided into the following stages:

- Collecting information
- Project positioning
- Project development
- Creation of the report
- Registration of the explanatory note
- Delivery of the project

In accordance with the functionality of our application, there are two types of scenarios for its use. These scenarios differ in the type of end-user. In the first case, the user is a consumer who can order some product. In the second case, the user is an administrator, who can operate with products, as well as regulate the user's workflow.

In the process of development, projects were assigned 2 roles:

– Consumer. In the process of the application usage consumers can get some information about the product with the ability to create an order, update an order, reserve some products or delete related data.

– Administrator. Can manage consumer's access to the application functions, update any product or configure already created orders.

The basic workflow can be divided into several parts. First a consumer can login or register in the system. After that, a consumer has the ability to check some available products. If a customer will find any needed product, then the order can be created.

Stepan Dubrovski

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

DEVELOPMENT OF A MOBILE CLIENT FOR A DISTRIBUTION NETWORK

In the development process, a variety of tools and technologies have been used. Java was chosen as the programming language. The choice fell on Java because it has been the main development language for Android for a while and has wide libraries, community and plugins support. Also, it gives a possibility to migrate to Kotlin language with no problem and big-time consumption. Additionally, at the moment it is one of the most popular programming languages. Java is an indispensable tool for developers and has opened up huge opportunities for them. Its community is growing and this allows it to develop stable and optimized applications with high tempo.

For UI layouts development XML standard is used. Extensible Markup Language defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts. Declaring UI in XML gives the possibility to separate the presentation of an app from the code that controls its behavior. Using XML files also makes it easy to provide different layouts for different screen sizes and orientations.

In addition, Room SQLite was used as a way to save app data at the local DB. The Room persistence library provides an abstraction layer over SQLite to allow fluent database access while harnessing the full power of SQLite. In particular, Room provides such benefits as compile-time verification of