8. Given:

Customer: Bookstore.com
Need to implement Java based web application where

1. an admin user can add a new book,
2. main index page can show all existing books.
    a. Could be organized in pages, but there should be visible at least 50 books per page
    b. Or you may choose additional content loading while user scrolls at bottom of page
3. Newest books are located at top of the page.
4. Main page load time cannot exceed 2 seconds.
5. System cannot store a new book which name already exists.
6. It is forecasted 10 parallel user sessions per second, during workdays from 11:00 to 15:00.

Delivery:
8.1. Please describe / draw main components of web application, how would you build the system. There are no strict requirements to UI, as well you may include minimum amount of data/fields to be requested from end-users and stored in DB. Please describe a reason of choosing an architecture, technologies, frameworks, etc. You may draw by hand and take a photo, e.g. its not required to use tools for UML, dataflow, MS Office, etc.
8.2. Describe risks (if any) you would see in your chosen solution.
8.3. Provide source code and pre-compiled package.
8.4. Provide “short” setup guide, e.g. what is target OS, what to install (appserver, DB, …), how to deploy application packages and how to run all services so the application would be up-and-running at local environment and accessible in browser.

As soon as you are ready, please upload all delivery (including un-finished & drafts) to Tieto in folder “release v.1.0.”

(Optional) 9. Given:

Prerequisites: completed task 8 with included source codes & installed environment
New change request to the Bookstore application.
A subcontractor is building an e-commerce system where one of sales channels is going to provide books for sale. It was decided that our Bookstore system will manage information about books including price, but the e-commerce system will take care of stocks and payments.
Please update the application so

1. it’s possible to add a price to a book
2. add some unit tests
3. create a REST webservice which provides a list of books: name & price
4. e-commerce system will synchronize this data every midnight at 00:00 EET
    a. if there is a new book name, this will auto-create a new product in sales channel, if its already exists, then this will update only a price to a product.
    b. You may recommend best solution for this integration

Delivery:
9.1. Provide updated source code and pre-compiled package.
9.2. If it is necessary, provide also updated the “short” setup guide.

As soon as you are ready, please upload all delivery (including un-finished & drafts) to Tieto in folder “release v.2.0.”

(Optional) 10. Given:

Prerequisites: n/a
Create a mock-service of the webservice which was implemented in task 9. The mock service should return only one book: name & price, e.g. “Alice in the wonderland”; “9.99”
Locate the mock-service in Docker container, so it would be fast to start up at any developer’s or tester’s host machine.
Alternatively, this task could be counted as completed if you wrap your environment of Task 8 or 9 into Docker.

As soon as you are ready, please upload zipped docker image to Tieto in folder “mock”.