PT-01 – Create New Book Record and Data Storage

1.1 – Creating buttons and book addition form on the GUI.

Description: Testing of all front-end components designed with Angular.
Pre-Conditions: Angular based front-end design must be established with the boot-strap and its connection to the Java back-end have to work correctly.
Post-Conditions: All buttons and form parts are expected to be ready to use.
Data Required: Random data samples that can come to certain parts.

1.2 - Filling all parts of the form completely.

Description: The test of filling all the information necessary to register a new book in the system in the right places without any exceptions.
Pre-Conditions: In order to get information from the user, all the necessary form must be created on the GUI.
Post-Conditions: When no data is entered in the required fields and is empty, the error stating that it should be filled is expected to be shown.
Data Required: Certain parts of the form are not filled in and try to add process.

1.3 – Form Validation.

Description: Performing all validation tests before creating a book to the system.
Pre-Conditions: Every time the value of a form control changes, Angular runs validation and generates either a list of validation errors, which results in an INVALID status, or null, which results in a VALID status.
Post-Conditions: Improve overall data quality by validating user input for accuracy and completeness.
Data Required: Inaccurate data in some parts, for instance wrong data in stock, price and date.

1.4 – Successful Data Storage .

Description: Testing data storage after clicked 'ADD' button
Pre-Conditions: All connections between front-end and back-end must be established and working properly.
Post-Conditions: The entered data is sent to the Java via http path, the book adding function is expected to work and the book has been added to the database and viewing in MySQL workbench.
Data Required: Newly entered book information data in json format.

PT-02 – Read Book Entities from Database and Display on GUI

2.1 - Receiving and sending data with Hibernate.
Description: Taking the created book model from the database via hibernate and sending it to the website path in json format with mapping.
Pre-Conditions: Book records have to already been created and must be stored in the database.
Post-Conditions: The data taken from the database in the desired directory must be displayed in json format.
Data Required: All book records stored in database.

2.2 – Creating tables to list records on the front-end.
Description: Designing to books listing section in the admin interface with angular so that the records can be listed properly and in the correct sections.
Pre-Conditions: Book listing component has been created with Angular.
Post-Conditions: All fields are ready before book records are received and they match correctly with the data.
Data Required: Sample test data in json format.

2.3 – Fetching and Displaying data in GUI.
Description: Reading the data on the front-end and placing and displaying it in the right places in the table.
Pre-Conditions: All fields to be listed data must be created.
Post-Conditions: Angular ngFor is an inbuilt directive that helps us to loop through the backend data and display it on the frontend. We need to display the data in whatever format we want.
Data Required: All book records stored in database.

2.4 – Fetching and Displaying data in GUI.
Description: Dynamically refreshing the table and page and showing current data after certain operations.
Pre-Conditions: Transmission of operations such as edit and delete after they are done.
Post-Conditions: Rendering Table Data with angular service. Displaying Data in the Angular HTML data table.
Data Required: Updated datasets.

PT-03 – Update One Book Record and Display Updated Version

3.1 - Using the edit button for update one record.
Description: Test of sending the id of the button's data after a data to be updated is determined
Pre-Conditions: The data to be updated must be specified on the GUI and click on the “edit” button next to it.
Post-Conditions: Determining the id of the data to be updated after clicking the button and sending it to the new page.
Data Required: One data record to be updated.

3.2 – Displaying information of selected record.
Description: Showing all information of the record to be updated on the following page.
Pre-Conditions: The page to be displayed must be created and edited before the ID and information of the required record is received.
Post-Conditions: It is expected that the information of the record that will be edited to all sections on the opened page will be brought completely.
Data Required: All information of the data to be updated.

3.3 – Updating Record.
Description: After all the information of the data is seen, only the desired parts are changed and edited.
Pre-Conditions: The parts of the data that are desired to be updated are shown and should be suitable for change.
Post-Conditions: All data information must appear in its new form after changes are made.
Data Required: The parts of the data to be updated.

3.4 - Form Validation when changing data.
Description: Performing all validation tests when updating a book to the system.
Pre-Conditions: Every time the value of a form control changes, Angular runs validation and generates either a list of validation errors, which results in an INVALID status, or null, which results in a VALID status.
Post-Conditions: It is expected that the file will be updated in accordance with all rules and necessary error parts will be shown.
Data Required: Inaccurate data in some parts, for instance wrong data in stock, price and date.

3.5 – Update function call and display updated record.
Description: Operation of the function required to complete the update process and checking the update.
Pre-Conditions: The update function has been created through hibernate and must be available at the service layer.
Post-Conditions: The edited record is sent to the back-end, calling the update function and completing the update. The current version of the record appears in the MySQL database.
Data Required: One data record to be updated.

PT-04 – Delete One Book Record

4.1 – Using the delete button for deletion.
Description: Test of sending the id of the button's data after a data to be deleted is determined.
Pre-Conditions: The data to be deleted must be specified on the GUI and click on the "delete" button next to it.
Post-Conditions: Determining the id of the data to be deleted after clicking the button and sending it to the new page.
Data Required: One data record to be deleted.

4.2 – Calling the delete function.
Description: The record to be deleted is sent to the back-end and calls the function.
Pre-Conditions: A deletion function properly written on the Java back-end.
Post-Conditions: The process of sending the data deletion query to the database through Hibernate.
Data Required: One data record to be deleted.

4.3 - Control of the record deleted in the database.
Description: Database check after record deletion.
Pre-Conditions: The record to be deleted must be in the database and there must be no obstacle for the record to be deleted.
Post-Conditions: After the deletion is completed, it is expected that the record is not available in the MySQL database.
Data Required: One data record to be deleted.