**Question 1:** Provide a way to get a product’s total number of customer reviews whose ratings are within a given range (inclusive).

**Solution Explanation:**

* Created a new interface: **custom/hybris/platfom/customerreview/dao/CustomerReviewPlusDao** extending CustomerReviewDao(OOTB) and declared: **public abstract int getTotalReviewsWithinRange(ProductModel paramProductModel, Double min, Double max)**.
* Created an implementation class: **custom/hybris/platfom/customerreview/dao/CustomerReviewPlusDaoImpl** extending DefaultCustomerReviewDao (OOTB), implementing the new interface: **CustomerReviewPlusDao**.
* With the new implementation class: **CustomerReviewPlusDaoImpl**, created a method - **public int getTotalReviewsWithinRange(ProductModel paramProductModel, Double min, Double max)** which uses Hybris Flexible Search to query ‘CustomerReview’ table against the product and also match the rating between the min and max values passed.
* Exposed the DAO method in the service class: **custom/hybris/platfom/customerreview/service/CustomerReviewServicePlus** under method name: **public Integer getTotalReviewsWithinRange(ProductModel product, Double min, Double max)**

**Question 2: Add the following additional checks before creating a customer review:**

**a. Your service should read a list of curse words. This list should not be defined in Java class.**

**b. Check if Customer’s comment contains any of these curse words. If it does, throw an exception with a message.**

**c. Check if the rating is not < 0. If it is < 0, throw an exception with a message.**

**If all the rules are passed, go ahead and create the customer review.**

**Solution Explanation:**

**To read the list of curse words:**

* Created method: **List<String> getCurseWords(LanguageModel language)** in **custom/hybris/platfom/customerreview/dao/CustomerReviewPlusDaoImpl.** This uses Hybris Flexible Search to create a query parameter to search the table ‘BlackListTerms’ against the matching parameter ‘language’ passed in the input, and returns the curse words as a List<String>
* Under service class: **custom/hybris/platfom/customerreview/services/CustomerReviewServicePlus**, used **‘LoadingCache’ – from Google Guava Cache** (Open source) to cache the curse words fetched from the data store.

**Overriding method createCustomerReview:**

* Created a service class: **custom/hybris/platfom/customerreview/services/CustomerReviewServicePlus** extending DefaultCustomerReviewService(OOTB).
* Within the service class, created **@Override createCustomerReview** **(Double rating, String headline, String comment, UserModel user, ProductModel product)** to customize the OOTB createCustomerReview method.
* Within the overridden method: Used StringUtils.split to break down the complete headline + comment and return the split words as a Set<String>
* Used retainAll collection utility to compare the list of curse words against the tokens generated from splitting the headline and comment.
* If retainAll returns any common occurrence with the curse words – throw InvalidDataException
* If input rating < 0 – throw InvalidArgumentException
* All custom validations needed for creating customer review are performed – so calling super.createCustomerReview() in the end.

**Question 3: Ensure the new functionality can be used elsewhere in the application (i.e. a bean containing the new functionality is defined within the customerreview-spring.xml).**

**Solution Explanation:**

* Created customerreview-spring-plus.xml.
* Created new alias with name ‘customerReviewServicePlus’ – whose bean class is mapped to the custom service class ‘custom.hybris.platfom.customerreview.services.CustomerReviewServicePlus’. Note: CustomerReviewServicePlus extends DefaultCustomerReviewService
* Created new alias with name ‘customerReviewPlusDao’ – whose bean class is mapped to the custom dao class ‘custom.hybris.platfom.customerreview.dao.CustomerReviewPlusDao’.

Note: customerReviewPlusDao extends customerReviewDao