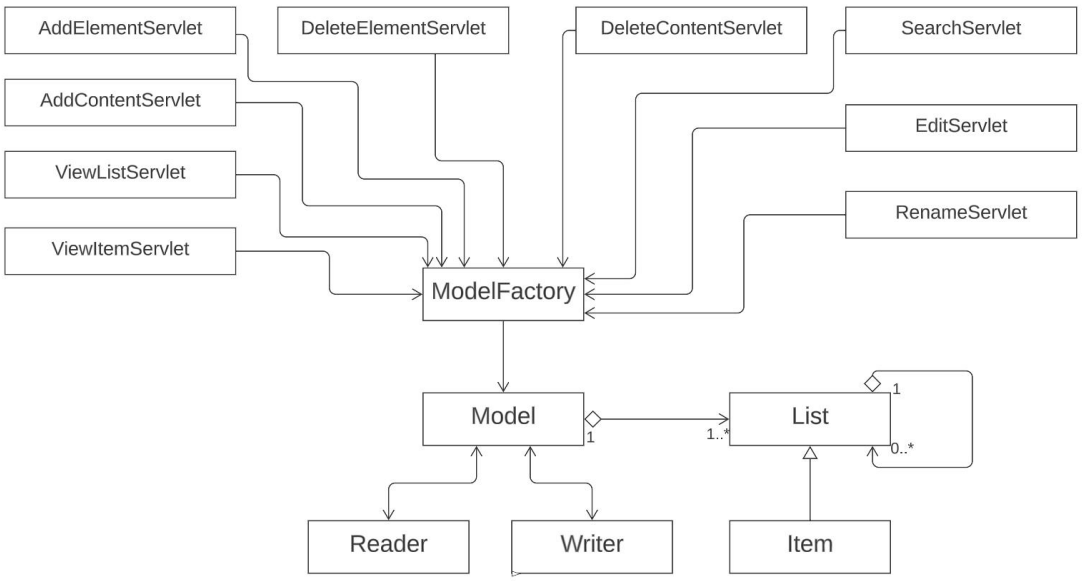
List Web Application Report

Features

My program implements the back end of the list web application, including functions for different operations of lists and items. All the back end functions needed for the 6 requirements are implemented. However, I have not managed to implement the web part of the application because I failed to write a valid JSP or a servlet class. Therefore, I only have the code for back end and a basic main web page.

UML Class Diagram



The lower part of this diagram shows the classes I actually implemented, while the upper part is based on my design for the application. Due to the failure when coding for JSPs and servlet classes, I was not able to realize my designs. If I have more time to figure out how JSP works, I would follow this design and finish my application.

Description and Evaluation

After reading the requirement 4, lists of lists, I decided to store both lists and items in one arraylist as the content for any particular list. Arraylist can only contain objects of the same type, so I decided to construct the Item class as the subclass of the List class. Then I could use the list keyword to store objects of both classes into one arraylist.

The Model class initializes a hashmap of id and list pairs, as well as a main list. The current maximum id is recorded by an instance variable. When a new list or item is created, a new id is taken, and the new id-list pair is stored in the hashmap. When a list or item is deleted, it will be excluded from the hashmap, and its parent list would remove it from the arraylist. After deletion, the maximum id needs to be updated to the existing maximum id stored in the hashmap.

Each list or item in the hashmap is written as one line in a csv file. ID, name, type (0 for lists and 1 for items), and parent ID are included. For items which have texts, urls or images as instance variables, 3 text files are created in the files directory to store corresponding contents. Each time before saving all the information to files, the current text files in the files directory need to be deleted to avoid disturbance.

My design includes 6 useful classes. List and Item classes are used to construct the two data structures, and Model class does the overall management. The ModelFactory class provides the servlet classes with an easy access to the appropriate model. Reader and Writer classes are responsible for loading and saving the data.

However, there are shortcomings in my design. For example, it may be better to construct an abstract class for List and Item classes to extend. Only the Item class needs instance variables for contents like texts or urls, so corresponding set and get methods are needed for Item class only. But since the List keyword is used at most places in the code, the List class has to have these instance variables and methods even if it does not need them, and the programmer needs to control carefully that only item objects use these methods. This can be considered as a bad design and should be improved.

Moreover, the code is messy now. Due to the lack of realization of servlet classes and JSPs, many methods implemented are not used. Redundant methods may be mixed in without notice. The code needs to be tidied up after completion.

The quality of List and Item classes needs to be improved, and the quality of other classes is better. Overall, the application needs to be completed and optimized further.