# Reflection

## Task 1 - Test plan

I have done tests before, but never a test plan so I don’t really know anything, really hard know what’s right or not. Should I be testing the small parts or what the user see? I decided to go with the use cases as that’s the only part I’ve made in this, and if use cases will fail (and not by the use case itself) I will know that something is wrong in the server. I don’t know what to with the dynamic, static tests as that’s more like a test design (as I got it). What I was aiming for was to get a clear picture on what should be tested (delete and getbooks), and by testing them I also test the system because if system isn’t working then use cases aren’t either.

## Task 2 - Test cases

I defined what should be tested and what to expect by the outcome, I did not include what to do with result as I can’t tell by the task if it should be an implementation or just a text. It does say create test cases but in task 3 it says that create so I will go for that (text in task 2 and implementation in task 3). The table will probably be modified in the future.

## Task 3

Just like the rest it was quite hard because it’s all new and there’s much information, what I mean is it is hard to know exactly what and how. I decided to test the two use cases remove and get books, remove book could be tested by checking the book list before and after to compare. But as for get books there wasn’t very much to test, either you get the books or you don’t, you could go in deeper and test on how well the xml management is done. But I don’t think I should test but only the things I made (I hope).

## Task 4

This has been re-written many times now, I first made a mistake thinking task 4 was task 3. But anyway task itself was not that hard, the hard part is to know if I’m doing it right or not (main concern is the task plan) but I tried to explain what needs to be tested and how so hopefully this is okay.

For this task I decided to test the response when calling the API for different request, they all failed when it came to 404 response. The problem was not in the recourse files but in the books.js file itself (API file). A 404 implementation (not found) were made and the test now checks out.