Dynamic tests

Dynamic tests are great for verifying correctness and overall improve the quality of the software. In tandem with static testing they are one of the main pillars of quality in the development phase of the application.  
In dynamic test we test the behavior of the software’s code, the software must be completed or be close to completion for it to be tested. By giving input values we check if we get the expected result. This is done by executing different test cases (done automatically via an automated process or done manually).

Test cases: are documents in the form of tables consisting of a set of variables and conditions under which the correctness of the software application is to be pre-determined in order to verify its functionality.  
Test scenario: is a test procedure which is a set of test cases built on the basis of business requirements. We put our selves in the shoes of the final users and go through the software to execute the test cases in different sections of the application (being the scenarios).

On top of that, in order to decide the input values for each of the test cases boundry value analysis. Therefore, based on the boundaries that each variable has, we decided to test values before, at (where possible), and right after the boundry.

We have included two different scenarios consisting of various test cases: Logging into the application and checking out a book

|  |  |
| --- | --- |
| Identifier | 16 |
| Description | Member wants to log into the application |
| Precondition | Must have a valid SSN, a password consisting of both numbers and letters with the minimum length of 6 characters and the maximum of 16 |
| Estimation | No estimate given |

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario: Log In – 16 | Valid test cases | | |
|  | SSN | Password | Expected result |
|  |  | 123abc | invalid |
|  | 1 | 123abc | invalid |
|  | 12345678 | 123abc | invalid |
|  | 123456789 | 123abc | valid |
|  | 1234567891 | 123abc | invalid |
|  | 12345679a | 123abc | invalid |
|  | 123456789 | 123 | invalid |
|  | 123456789 | abc | invalid |
|  | 123456789 |  | invalid |
|  |  |  | invalid |
|  | 123456789 | 1a | invalid |
|  | 123456789 | 123456789abcdefghi | invalid |

|  |  |
| --- | --- |
| Identifier | 17 |
| Description | Member wants to checkout a book |
| Precondition | Book must have a valid Copy ID, from date (date of checkout), member must have a number of books borrowed lower than 5 (but not null) and a valid SSN |
| Estimation | 05:00:00 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Scenario: Checkout - 17 | Valid test cases | | | | |
|  | Copy ID | SSN | From date | Number of books borrowed | Expected result |
| 1. |  |  |  |  | invalid |
| 2. |  | 12345678 | 12-12-12 | 0 | invalid |
| 3. |  | 1233456789 | 12-12 | 1 | invalid |
| 4. |  | 123456789 | 12-12-12 | 4 | invalid |
| 5. |  | 123456789 | 12-12-12 | 5 | invalid |
| 6. |  | 123456789 | 12-12-12 | 6 | invalid |
| 7. | 1 | 12345678 | 12-12 | 4 | invalid |
| 8. | 1 | 1233456789 | 12-12-12 | 5 | invalid |
| 9. | 1 | 123456789 | 12-12-12 | 6 | invalid |
| 10. | 1 | 123456789 | 12-12-12 |  | invalid |
| 11. | 1 | 123456789 |  | 0 | invalid |
| 12. | 1 |  | 12-12-12 | 1 | invalid |
| 13. | 1 | 123456789 | 12-12-12 |  | invalid |
| 14. | 1 | 123456789 | 12-12-12 | 0 | valid |
| 15. | 1 | 123456789 |  | 1 | invalid |
| 16. | 1 | 123456789 | 12-12-12 | 4 | valid |
| 17. | 1 |  | 12-12 | 5 | invalid |
| 18. | 1 | 12345678 | 12-12-12 | 6 | invalid |
| 19. | 1 | 123456789 |  | 4 | invalid |
| 20. | 1 | 123456789 | 12-12-12 | 5 | invalid |
| 21. | 1 | 123456789 | 12-12 | 6 | invalid |
| 22. | 1 |  | 12-12-12 |  | invalid |
| 23. | 1 | 12345678 | 12-12-12 | 0 | invalid |
| 24. | 1 | 1233456789 | 12-12-12 | 1 | invalid |
| 25. | 1 | 123456789 | 12-12 |  | invalid |
| 26. | 1 | 123456789 | 12-12-12 | 0 | invalid |
| 27. | 1 |  | 12-12-12 | 1 | invalid |
| 28. | 1 | 12345678 | 12-12-12 | 4 | invalid |
| 29. | 1 | 1233456789 |  | 5 | invalid |
| 30. | 1 | 123456789 | 12-12-12 | 6 | invalid |
| 31. | 1 | 123456789 | 12-12-12 | 4 | invalid |
| 32. | 1 |  | 12-12-12 | 5 | invalid |
| 33. | 1 | 12345678 |  | 6 | invalid |
| 34. | 1 | 1233456789 | 12-12-12 |  | invalid |
| 35. | 1 | 123456789 | 12-12 | 0 | invalid |
| 36. | 1 | 123456789 | 12-12-12 | 1 | invalid |

As seen in the XXX table, there are 36 possible test cases. Although this is a very achievable plan, in order to save time (both writing and execution), we have decided to reduce the number of scenarios by doing some equivalence partitioning. In order to do this, we have used <https://pairwise.teremokgames.com/>, an online tool that generates the smallest number of scenarios.