DApp Design Decisions

1. We needed tokens that could point to a particular file, so ERC20 was out of the picture. NFT was the solution.

2. Decided to settle on ERC721 standard, and started to work on it. We tried to build the methods from scratch, and then tried to build on top of open-zeppelin when we realized the amount of work.

3. We switched back to own implementation when open-zeppelin's code was too heavy for us, where a lot of methods were not needed for our specific application.

4. In the code itself, arrays and loops etc. were completely avoided to minimize gas cost. Mapping is extensively used because of its ease of lookup etc.

5. Each token corresponds to each individual copy of the book.

6. A list of tokenIDs owned by a user is not stored in a list anywhere, since that would require a dynamic array. The user will have to save the IDs of the tokens elsewhere for later use. (This would have to be implemented in the real world, of course).

7. A front-end validation forces the publisher to set the sale Commission to be less than 50%. This is to nudge the publisher away from having the bigger piece of the pie at every transaction.

8. No token can be burned (destroyed) by the owner.

9. There is no approval functionality in the ĐApp, because that would give more than one person permission to access the token data at any given point in time. This would break the digital scarcity assumption. (Two people shouldn't be able to read one book simultaneously)

10. The front-end is a single page, where each box corresponds to each view and the behavior therein. ReactJS is used for front-end.

One of the main objectives of the project was to implement a renting option, where a token owner can rent his token and thereby the token data to any third party at a price of the owner's choice. The owner won't have access to tokenData whenever the token is rented out (just like renting a book in the physical world). This would mean that one can generate revenue off of the digital content one owns, but on a global scale. But this functionality was dropped due to time constraints.