1. **(10pts) How would you review the initial requirements and specification? (Provide details.)**

   I am using the Naïve approach to read the specification carefully, then break down the specification into functional requirements and non-functional requirements. The process will let you have a full understanding of the specification of the system. It is suitable for only reading and planning specifications but not ideal for code reviews because the information is vast and time-consuming. But in the real world, active specification is a better method in this case, and it consists of five steps process:

   1. Prepare the documentation for review
   2. Identify the specialized reviews
   3. Identify the reviewers needed
   4. Design the questionaries
   5. Conduct the review

   It gives more opportunities for a one-on-one discussion between the reviewer and the specification team, making it easier for the team to speak up and reduce errors when approaching.

   Functional requirements describe the service of the software or its component.

   For customers:

   a. Clients need the abilities for create their accounts, to store payments, and re-access e-content for previous purchased.
   b. The online eBook store has the abilities let clients to purchase both eBooks and physical books. All of the physical books are printed by specialize machine.
   c. Create new payment method or access previously saved payment method. All saved payment should be non-expired.
   d. Customers' needs to display the identity as "customer" on the site.

   For authors:

   a. Authors needs to show the identity as authors status on the site.
   b. Able to monitor of their book sales activity.

   For retailer:

   a. Retailer owns the administrators and all authority of the site.
   b. Administrators can distribute the role to the user.
   c. Administrators can control of the promotions, such as create a holiday sale, manage the coupons, and discontinue/delete the promotions
   d. All customer's sale taxes need to be charged based on the address (state, city, or zip code)

   Non-functional requirements describe the limits of a system's functionality and its behavior.

   For customers:

   a. Account login – allow account members to access the site by their username and password. After verified the users, they will be able to purchase books online or update their personal information.

b. Account logout – allow members to logout their account.
c. Account registration – allow users to register new username and password as a new user.
d. User status – display member status as customers and give limited function.
e. Update account information – allow users to edit their stored information.
f. Comment – allow members to leave the comment about books their purchased and certified they had purchased the book. This function can provide the feedback to the author and review of the book.
g. Description – this function show the intro, catalog, author, title, ID number, and ISBN number of the book.
h. Add to cart – allow users to store the book they want to purchase.
i. Delete from cart – allow users to remove any unwanted book s
j. Search – allow users to search the eBooks or physical books by title, ISBN number, category, or author.
k. Checkout – allow account users purchase books securely. After the purchased success, the system will provide a tracking number of the order to the member.
l. Purchase history – allows member to view previous purchases or invoices.
m. Customer service – to provide the troubleshooting service to the customers by phone number and email address.
n. Shipping status – allow users to view the status of the shipment and the tracking number from the purchased order.
o. Help – guide and introduce all the functions to the new user.

For authors:

a. Account login – allow authors to access the site by their username and password. After verified the authors, they will be able to purchase books online or update their personal information.
b. Account logout – allow authors to logout their account.
c. Account registration – allow authors to register new username and password as a new user.
d. User status – display member status as “author” and give limited function.
e. Update account information – allow authors to edit their stored information.
f. Sales activity – provide sales activity to the author. How much books they sales and the prediction of the trends (what customers favorite book are).

For administrators:

a. Create role to the user – administrators can create role and authority to individual user.
b. Control of the promotions – create, manage, and discontinue or delete the promotions.

For retailer:

a. Sale tax based on different areas – allows to adjust the sale tax based on the account members address (US states, city, or zip code).

2. **(20pts) You cannot test the payment system by way of the bank on a frequent basis, because it would be slow, and the bank does not want to be a heavily-used part of your testing infrastructure. (Banks do offer “test servers” to send fake payment requests to in order to ensure you can actually talk to a real server, but this is still too slow for unit tests.) How would you instead test payment processing logic, without relying on the bank? Similarly, how would you test print-on-demand logic, without wasting expensive ink and paper and requiring a human to read the results?**

   To test the payment processing logic without relying on the bank, developer can use open box testing to validate the logic of the application that implemented by the code. The code coverage can check the number of lines is executed by tests and branch coverage can make sure all reachable code is executed. Developer can create an external party’s sandbox (Chapter 18 in the textbook) or test environment to fake the system to interact with. All the configuration can mirror as a real bank environment. All the actions and the change does not affect the system because it is a virtual environment for testing before approach to the production phase.

   To avoid wasting expensive ink and paper, I will use control flow testing to ensure the order's condition. Each book has its program paths. When a customer orders a physical book, the backend will notice the printer to print that book. The testing will take the book's ISBN and then verify the author, title, and the years of the book. Also, checking that the customers had paid for the book before the book start printing to prevent wasting.

3. **(20pts) Closed-box and open-box testing approaches are different, but complementary. Please describe the trade-offs, and give examples of things each approach is good for finding that the other approach is poorly suited to finding. (Recall that open-box testing includes code coverage, but also more than just code coverage.) Given these trade-offs, are there parts of the system that really should have a heavier emphasis on one type of testing rather than the other? Specifically, identify areas where closed-box testing would be especially important/appropriate, and places open-box testing would be especially important/appropriate.**

   The closed-box testing is used to check what functionality of the system under test. A tester doesn't know any information about the internal structure of the system. It is a high level of testing that concentrates on the software's behavior and mainly focuses on external or end-user perspective. For instance, all pair testing is one of the critical types of the closed box. Generally, most common bugs in a program are triggered by an interaction between pairs of parameters. All pair testing can arrange variables and create a test suite. Thus, the pairwise testing technique can dramatically reduce the number of combinations to be covered but remains very useful in fault detection. The black box testing is suitable for large code segments and applies to acceptance and system testing. However, it requires testers to write many test cases to check the condition that could be tested by only one test case, and the tester cannot see the parts of the codes for troubleshooting.

   The open-box testing checks the quality of the code. Testing is based on coverage of code statements, branches, paths, or conditions. The open-boxing testing method assumes that the path of the logic in a program is known. Testers knew the internal structure of the program. It focused on code structure, conditions, paths, and branches. For instance, Branch coverage is also known as Decision coverage or all-edges coverage. It covers both the true and false conditions, unlikely the statement coverage. The decision coverage will compare completely two or more variables, then to get 100% statement coverage. And also, it will give decision coverage as well. The 100% decision coverage is Ture or False condition. Thus, the final set goal has two test cases for 100% decision coverage. The white box testing can optimize the code and let the software keep track of their testing. On the downside, it takes a long time to perform the testing, and it is challenging to keep track of which part of the code is tested. Also, it needs specialized tools (debugging tools and analyzers) to maintain the.

4. **(10pts) Customers and authors alike must both be able to log in and access relevant information. Customers can view their past orders, re-download previous ebook purchases, and change payment information. Authors should be able to log in, access statistics on how their books are selling. Store administrators can create and disable coupon codes. Granting or denying access to these pieces of functionality incorrectly would be a major issue. Give 6 tests you would write for checking user access to different functionality.**
    1. Regression testing – to ensure after customers make change (update payment, username, password, or personal information), the system can still function normally.
    2. Unit testing ensures the customers, authors, and administrators' functionality meets as intended, such as register account, log in, view order, create coupon codes, etc.…
    3. Performance testing – to check the website's speed, response time, reliability, and stability under a specific workload. It also determines how many users the website can handle.
    4. Security testing – to inspect the threats in the system and measure the potential vulnerabilities of the website. The testing can manage user access correctly like the administrator had the website's highest authority, and customers can perform specific actions (buy books, check order).
    5. Cross-browser testing – making sure the website's layout is expected in different browsers and different devices (smartphone).
    6. Usability – to test the website with real users (customers and authors). Give them a list of tasks and observed if they encounter problems or confusion.

5. **(10pts) For the first version of the software, usability is important, but you have a limited usability budget in both time and funds: you can pay 6 prospective users (few) for one hour each (relatively little time) to participate in usability studies. What kinds of studies do you pursue? What kind of users would you find for these? Justify your choices.**

    During the usability testing, I'm pursuing the studies of:
    1. Learnability – make the website user friendly.
    2. Efficiency – let the users complete their tasks quickly and effectively.
    3. Memorability – do not require to memorize anything about how to use the website to purchase books.
    4. Errors – to minimize the problems a user may encounter while using it.
    5. Satisfaction – To extract user satisfaction and experience for improvement.

    I would be looking for the actual user who uses a similar website or who had experienced purchasing online or selling books online. The goal is to understand how real users interact with the website and giving evaluations to improve the product based on the results.

6. **(5pts) What kinds of accessibility testing should you perform on the web interface? How (at a high level) might you implement some of these via something like Selenium, and are there other kinds you might need to check manually?**

   A web interface should ensure testing for users with disabilities and include accessibility features, such as visual disability, insufficient vision disability, and visual disability. Selenium web driver is a GUI testing tool for web applications and is used to take any browser to specific pages. It finds and checks the UI elements with DOM lookups. Selenium can automate test web applications, but it can't see CSS's effects and the visual layout. Therefore, those parts required a test manually.

7. **(10pts) What is equivalence partitioning (examples are insufficient, but can add clarity), and what are some examples of useful places to apply equivalence testing specifically that might arise in a project like this?**

   Equivalence partition is a test case in black-box testing. It intends to reduce the total number of test cases by dividing the input data into different equivalence classes. It does cover the maximum requirements with a finite set of testable test cases. For instance, equivalence partition can apply to the book, author, or customer name that is alphabetic because the name cannot be numeric values. Or ISBN, each book has an individual and unique number for its identity. The ISBN can never be alphabetic or decimal.

8. **(10pts) What kinds of performance-related testing would you do for a system like this, and why?**

   I will perform the following performance test:

   1. Load Testing – to ensures that the website performs as expected in production. This testing can track where and when your website breaks, and you can have time to fix it before you approach production.
   2. Stress Testing – this test can put your load to its limit. It helps find where your website's boundaries are for acceptable performance and the upper limits of the website's capacity using a load beyond the expected maximum.
   3. Longevity Testing – to ensure website stability and serviceability over a more extended period. It can make sure how the website would against appropriate load and stress conditions with real-time traffic.
   4. Capacity Testing - determines how many users or transactions your website can handle before it reaches its limit. It is an excellent way to avoid potential problems in the future.

9. **(5pts) Identify one part of the course material you feel you wish either could have been covered in more depth (and a high-level why) or a topic you feel perhaps should have been in the course but was not. This question will not be graded on the details of your choice(s) for the question, but will be used to improve the course in subsequent years. If you answer the question seriously, you will receive the 5 points.**

   I hope the course can cover mobile testing such as demonstrate the mobile app layout and how to create a mobile testing plan. I think mobile testing is crucial because it is against a mobile application to ensure its usability and functionality. We use the apps on our phones every day, and sometimes you can find out that some applications do not support some of the operating systems. Mobile testing can have better UI/UX and make a five start the mobile app to maintain your user.