

- **Test 1: Books page**

- The user acceptance test cases will include making the correct calls from the API in order to correctly populate the cards on the screen.
- Since the API doesn't contain images, we will have to create a table to populate that ourselves.
- Therefore each of the cards will contain images, books name, author name, and the genre.
- We will need to break this into 2 units because of the different implementations – the API and the database.
- The API unit test will be to test if the information regarding the book name, author, and genre are being returned correctly.
- The database unit test will be to make sure if the images are being returned correctly.
- The test case will pass if the frontend is able to successfully make calls to each individual book in the API, AND, if it is able to match the calls to the database.

- **Test 2: Single Books page**

- The single books page will contain the text of the book that was selected by the user.
- It will also have cards on the bottom of the page which showcase any annotations that people may have made.
- It will also have an annotations button at the bottom of the page which will lead to the annotations page.
- It will make calls to the database to get this information so it is displayed.
- The unit test cases in this will be separated into database calls, and frontend calls.
 - The database calls, depending on the implementation of our tables, might be separated into different unit test cases.
 - However, the main underlying principle will be the requirement of getting the information from the tables and displaying them onto the frontend.
 - The annotations unit case will be pretty simple as all it needs to do is make sure that it can go to the annotations page, or open another annotation.
- The database will have to select the correct information based on the API call made from the user.
- The test case will pass if the database is successfully able to return the data from the API calls.

- **Test 3: Annotations Page**

- This page will allow users to edit their previous annotations, or make new annotations.
- These annotations include being able to highlight text, add text (annotations) based on what is highlighted, and write a review of your thoughts.

- This will be tested through many backend and database calls as we have to save each annotation in the database and be able to get that information.
- The unit test cases will be split into each of the different features created in the frontend, (highlight, annotation). Also will be split into backend and database calls in order to make sure that what is annotated is able to be sent to the database, and if we are able to see what others wrote.
- The test case will pass if we are able to get all the annotations made by other people onto the annotations page, and if we will be able to save our own annotations into the database.