**TEAM – KIPP**

**Members - Kishor, Prasanna, Prabhas, Preetish, Meghan, Puneeth**

**TEST CASES**

**Test Cases for Searching for a Book:**

**Method - def search\_for\_book(book\_name, isbn=None)**

Test Case 1: Provide a valid book\_name that exists in the database.

Result – Pass

Test Case 2: Provide a book\_name that does not exist in the database.

Result – Pass (Taken care by python exception handling)

Test Case 3: Don't provide a book\_name parameter.

Result – Pass (book\_name takes default parameter ‘None’ if nothing is provided, so no error when it is empty)

Test Case 4: Provide a valid isbn that exists in the database.

Result – Pass

Test Case 5: Provide an isbn that does not exist in the database.

Result – Pass (Taken care by python exception handling)

Test Case 6: Don't provide an isbn parameter.

Result – Pass (isbn takes default parameter ‘None’ if nothing is provided, so no error when it is empty)

Test Case 7: Provide both a valid book\_name and a valid isbn that exist in the database.

Result – Pass (Any one parameter is needed to search a book)

Test Case 8: Provide both a valid book\_name and an invalid isbn.

Result - Pass (Any one parameter is needed to search a book)

Test Case 9: Provide an invalid book\_name and a valid isbn.

Result - Pass (Any one parameter is needed to search a book)

Test Case 10: Provide both invalid book\_name and isbn.

Result – Pass (Taken care by Python Exception Handling).

**Test Cases for Issuing a Book:**

**Method - issue\_item(student\_name,book\_name,due\_date)**

Test Case 1: Provide valid student\_name, book\_name, and due\_date.

Result – Pass

Test Case 2: Provide an empty student\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 3: Provide an empty book\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 4: Provide an invalid due\_date format.

Result – Pass (Taken care by Python Exception Handling)

Test Case 5: Provide a student\_name that doesn't exist in the database.

Result – Pass (Taken care by Python Exception Handling)

Test Case 6: Provide a book\_name that doesn't exist in the database.

Result – Pass (Intimated to the User)

Test Case 7: Try issuing the same book to the same student twice without returning it in between.

Result – Fail (We Have to implement this)

Test Case 8: Provide a valid student\_name and book\_name, but no due\_date.

Result – Pass (Taken care by Python Exception Handling)

Test Case 9: Provide a valid student\_name and due\_date, but no book\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 10: Don't provide any parameters (empty method call).

Result – Pass (Taken care by Python Exception Handling)

**Test Cases for Collecting Back the Book**

**Method - collect\_item(member\_name,issue\_id)**

Test Case 1: Provide valid member\_name and issue\_id for a book that was issued to that member

Result – Pass

Test Case 2: Provide an empty member\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 3: Provide an empty or invalid issue\_id.

Result – Pass (Taken care by Python Exception Handling)

Test Case 4: Provide a member\_name that doesn't exist in the database.

Result – Pass (Taken care by Python Exception Handling)

Test Case 5: Provide an issue\_id that doesn't exist in the database.

Result – Pass (Taken care by Python Exception Handling)

Test Case 6: Collect the book after the due date to check how the system handles late returns.

Result – Pass (Taken care by Trigger in the Database)

Test Case 7: Don't provide any parameters (empty method call).

Result – Result – Pass (Taken care by Python Exception Handling)

**Test Cases for Adding Rating and Reviews for a Book**

**Method - get\_rating\_and\_review(member\_name,rating,review)**

Test Case 1: Provide valid member\_name, rating, and review.

Result – Pass

Test Case 2: Provide an empty member\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 3: Provide an empty review.

Result – Fail (Not Implemented)

Test Case 4: Provide an invalid rating (e.g., above the maximum allowed value or below the minimum).

Result – Fail (Not Implemented)

Test Case 5: Don't provide any parameters (empty method call).

Result – Pass (Taken care by Python Exception Handling)

**Test Cases for Edit\_number\_of\_copies**

**Method - edit\_copies(book\_name,action,no\_of\_copies)**

Test Case 1: Increase the number of copies (action = 'increase') for a valid book\_name by providing a valid no\_of\_copies.

Result – Pass

Test Case 2: Decrease the number of copies (action = 'decrease') for a valid book\_name by providing a valid no\_of\_copies.

Result – Pass

Test Case 3: Provide an empty book\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 4: Provide an invalid action (e.g., something other than 'increase' or 'decrease').

Result – Pass (Taken care by Python Exception Handling)

Test Case 5: Provide an invalid no\_of\_copies (e.g., a negative value or non-integer value).

Result – Fail (Not Implemented yet)

Test Case 6: Provide a book\_name that doesn't exist in the database.

Result – Pass (Taken care by Python Exception Handling)

Test Case 11: Don't provide any parameters (empty method call).

Result – Pass (Taken care by Python Exception Handling)

**Test Cases for Adding a New Book**

**Method - add\_new\_book(new\_book\_isbn,new\_book\_name,new\_book\_author\_name,new\_book\_vendor\_name,new\_book\_publisher\_name,new\_book\_no\_of\_copies=1)**

Test Case 1: Add a new book with all required parameters (ISBN, name, author name, vendor name, publisher name).

Result – Pass

Test Case 2: Add a new book with all parameters including new\_book\_no\_of\_copies.

Result – Pass

Test Case 3: Provide an empty new\_book\_isbn.

Result – Pass (Taken care by Python Exception Handling)

Test Case 4: Provide an empty new\_book\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 5: Provide an empty new\_book\_author\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 6: Provide an empty new\_book\_vendor\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 7: Provide an empty new\_book\_publisher\_name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 8: Provide an invalid new\_book\_no\_of\_copies (e.g., a negative value or non-integer value).

Result – Fail (Not implemented)

**Test Cases for Recommending a Book**

**Method - recommend\_a\_book()**

Test Case 1: Provide a new book name, author(s), and publisher that do not exist in the recommendations or the library.

Result – Pass

Test Case 2: Provide a new book name and author(s) that exist in the recommendations but not in the library.

Result – Pass

Test Case 3: Provide a new book name that exists in the library but not in the recommendations.

Result – Pass

Test Case 4: Provide an empty book name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 5: Provide an empty author(s) field.

Result – Pass (Taken care by Python Exception Handling)

Test Case 6: Provide an empty publisher name.

Result – Pass (Taken care by Python Exception Handling)

Test Case 7: Recommend a book that already exists in the library.

Result – Pass

Test Case 8: Don't provide any parameters (empty input).

Result – Pass (Taken care by Python Exception Handling)