```
import pandas as pd

class BookLover:

    def __init__(self, name, email, fav_genre,
                 num_books=0, book_list=pd.DataFrame({'book_name':[], 'book_rating':
[]})):
        self.name = name
        self.email = email
        self.fav_genre = fav_genre
        self.num_books = num_books
        self.book_list = book_list

    def add_book(self, book_name, rating):
        if book_name not in self.book_list.values:
            new_book = pd.DataFrame({'book_name':[book_name],
                                     'book_rating': [rating]})
            self.book_list = pd.concat([self.book_list, new_book],
                                       ignore_index=True)
            self.num_books += 1
        else:
            return f"{book_name} is already in the book list for {self.name}"

    def has_read(self, book_name):
        if book_name in self.book_list['book_name'].tolist():
            return True
        else:
            return False

    def num_books_read(self):
        return self.num_books

    def fav_books(self):
        fav_books = self.book_list.loc[self.book_list['book_rating'] > 3]
        return fav_books
         1      from booklover import BookLover
import unittest

class BookLoverTestSuite(unittest.TestCase):

    def test_1_add_book(self):
        book_lover1 = BookLover('Hilde', 'ksg8xy@virginia.edu', 'romance')
        book_lover1.add_book('Dune', 5)
        books = book_lover1.book_list['book_name'].tolist()
        actual = 'Dune' in books
        message = "Book was not added successfully"
        self.assertTrue(actual, message)

    def test_2_add_book(self):
        book_lover2 = BookLover('Hilde', 'ksg8xy@virginia.edu', 'romance')
        book_lover2.add_book('Dune', 5)
        book_lover2.add_book('Dune', 5)
        books = book_lover2.book_list['book_name'].tolist()
        actual = sum([True for book in books if book == 'Dune'])
        message = "Same book was added more than once."
        self.assertLess(actual, 2)

    def test_3_has_read(self):
        book_lover3 = BookLover('Hilde', 'ksg8xy@virginia.edu', 'romance')
        book_lover3.add_book('Dune', 5)
        books = book_lover3.book_list['book_name'].tolist()
        actual = 'Dune' in books
        message = "Book not found in book list."
        self.assertTrue(actual, message)
```

```
30
31      def test_4_has_read(self):
32          book_lover4 = BookLover('Hilde', 'ksg8xy@virginia.edu', 'romance')
33          books = book_lover4.book_list['book_name'].tolist()
34          actual = 'Dune' in books
35          message = "Unread book found in book list."
36          self.assertFalse(actual, message)
37
38      def test_5_num_book_read(self):
39          book_lover5 = BookLover('Hilde', 'ksg8xy@virginia.edu', 'romance')
40          book_lover5.add_book('Dune', 5)
41          book_lover5.add_book('Emma', 5)
42          book_lover5.add_book('The Great Gatsby', 5)
43          expected = 3
44          actual = book_lover5.num_books
45          message = "Number of books in list does not match num_books."
46          self.assertEqual(expected, actual, message)
47
48      def test_6_fav_books(self):
49          book_lover6 = BookLover('Hilde', 'ksg8xy@virginia.edu', 'romance')
50          book_lover6.add_book('Dune', 5)
51          book_lover6.add_book('Emma', 3)
52          book_lover6.add_book('The Great Gatsby', 4)
53          book_ratings = book_lover6.fav_books()['book_rating'].tolist()
54          expected = 2
55          actual = sum([True for rating in book_ratings if rating > 3])
56          message = "Books in fav_books do not have ratings > 3"
57          self.assertEqual(expected, actual)
58
59  if __name__ == '__main__':
60      unittest.main(verbosity=3)
61
 1  test_1_add_book (__main__.BookLoverTestSuite.test_1_add_book) ... ok
 2  test_2_add_book (__main__.BookLoverTestSuite.test_2_add_book) ... ok
 3  test_3_has_read (__main__.BookLoverTestSuite.test_3_has_read) ... ok
 4  test_4_has_read (__main__.BookLoverTestSuite.test_4_has_read) ... ok
 5  test_5_num_book_read (__main__.BookLoverTestSuite.test_5_num_book_read) ... ok
 6  test_6_fav_books (__main__.BookLoverTestSuite.test_6_fav_books) ... ok
 7
 8  ----------------------------------------------------------------------
 9  Ran 6 tests in 0.004s
10
11  OK
```