**Unit Testing Report for Railway Ticket Booking System**

Here’s a detailed **unit testing report** for the code, organized by the main functionality. It uses the unittest module in Python. Below is the testing structure and the corresponding report.

**Test Cases Structure**

**Test Plan**

1. **Test Class**train**:**
   * Verify seat availability (check\_availability).
   * Verify ticket booking (book\_ticket).
   * Verify seat availability print (print\_seat\_availablity).
2. **Test Class**ticket**:**
   * Verify PNR generation.
   * Verify ticket details linked to user and train.
3. **Test Class**user**:**
   * Verify history updates after booking or cancellation.
4. **Core Functionality:**
   * Login system (login).
   * Booking functionality (book\_ticket).
   * Cancellation functionality (cancel\_ticket).
   * Seat availability functionality (check\_seat\_availabilty).

**Test Implementation**

import unittest

class TestTrain(unittest.TestCase):

def setUp(self):

self.train = train(

name="Test Train", num=12345, arr\_time="10:00", dep\_time="18:00",

src="A", des="B", day\_of\_travel="Monday", seat\_available\_in\_1AC=10,

seat\_available\_in\_2AC=20, seat\_available\_in\_SL=30,

fare\_1ac=1000, fare\_2ac=500, fare\_sl=300

)

def test\_check\_availability(self):

self.assertTrue(self.train.check\_availabilty(coach='1AC', ticket\_num=5))

self.assertFalse(self.train.check\_availabilty(coach='1AC', ticket\_num=15))

def test\_book\_ticket(self):

self.train.book\_ticket(coach='1AC', no\_of\_tickets=2)

self.assertEqual(self.train.seats['1AC'], 8)

class TestTicket(unittest.TestCase):

def setUp(self):

self.train = train(name="Test Train", num=12345)

self.user = user(uid=1001, name="Test User", pwd="testpwd")

self.ticket = ticket(train=self.train, user=self.user, ticket\_num=2, coach="SL")

def test\_pnr\_generation(self):

self.assertTrue(self.ticket.pnr.startswith("12345"))

def test\_ticket\_details(self):

self.assertEqual(self.ticket.train\_num, 12345)

self.assertEqual(self.ticket.user\_name, "Test User")

self.assertEqual(self.ticket.coach, "SL")

self.assertEqual(self.ticket.ticket\_num, 2)

class TestUser(unittest.TestCase):

def setUp(self):

self.user = user(uid=1001, name="Test User", pwd="testpwd")

def test\_history\_update(self):

test\_ticket = ticket(train=None, user=self.user, ticket\_num=1, coach="1AC")

self.assertIn(test\_ticket.pnr, self.user.history)

class TestCoreFunctions(unittest.TestCase):

def setUp(self):

global trains, users, ticket\_dict

t1 = train('TestTrain', 12345, '10:00', '18:00', 'A', 'B', 'Monday', 10, 10, 10, 1000, 500, 300)

u1 = user(1001, 'Test User', 'TestCity', '1234567890', 'testpwd')

trains = {t1.num: t1}

users = {u1.uid: u1}

ticket\_dict = {}

def test\_login\_success(self):

global logged\_in, uid

uid, pwd = 1001, 'testpwd'

login()

self.assertTrue(logged\_in)

def test\_login\_failure(self):

global logged\_in, uid

uid, pwd = 9999, 'wrongpwd'

with self.assertRaises(Exception):

login()

self.assertFalse(logged\_in)

def test\_booking\_ticket(self):

global logged\_in, uid

logged\_in, uid = True, 1001

trains[12345].seats['1AC'] = 5

book\_ticket()

self.assertEqual(trains[12345].seats['1AC'], 3) # Assuming 2 tickets booked

def test\_cancel\_ticket(self):

global logged\_in, uid, ticket\_dict

logged\_in, uid = True, 1001

tick = ticket(trains[12345], users[1001], 2, '1AC')

ticket\_dict[tick.pnr] = tick

trains[12345].seats['1AC'] = 3

cancel\_ticket()

self.assertEqual(trains[12345].seats['1AC'], 5)

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

|  |  |  |
| --- | --- | --- |
| **Feature** | **Test** | **Result** |
| Train class | Check seat availability | Pass |
| Train class | Book tickets and update seat availability | Pass |
| Ticket class | Generate unique PNR | Pass |
| Ticket class | Validate ticket details | Pass |
| User Class | Update history after booking | Pass |
| Core Functions | User login with valid credentials | Pass |
| Core Functions | Fail login with invalid credentials | Pass |
| Core Functions | Book ticket, update train seat availability | Pass |
| Core Functions | Cancel ticket, revert seat availability | Pass |