TEST PLAN

- Akshat Sharma (19HS20053)

Unit Testing Plan:

Class Railways:

- Validity is checked.
- The Methods likeGetDistance are tested.
- The friend function overloaded << operator is tested.
- Every output is compared to its Golden Value.

Class Station:

- First Construction is tested.
- The Methods like GetName, GetDistance are tested.
- The friend function overloaded << operator is tested.
- Every output is compared to its Golden Value.

Class Date:

- First Construction and Validity is tested.
- The Methods like getDay, getMonth, getYear, ComputeAge are tested.
- Exception Handling is tested
- The friend function overloaded << operator is tested.
- Every output is compared to its Golden Value.

Class Passenger:

- Good and bad passenger objects are constructed.
- We can see that the good passenger gets printed (using << overloaded operator) whereas the bad passenger gives an error message also

Class BookingClass and Hierarchy:

- Testing << operator. This operator calls methods like GetName, IsSitting, IsAC, GetNumberOfTiers, and IsLuxury methods
- if the printed values match golden values, it works fine.
- Testing virtual GetLoadFactor Method: calling GetLoadFactor of every final concrete class and comparing them to golden values. If no error prints on screen, the method works fine
- Every inherited class object is called and printed.

Class BookingCategory and Hierarchy(Categories) :

- The whole hierarchy consists of singleton classes.
- Unit testing checking whether every singleton is properly implemented or not.
- Each instance is constructed and the address is printed

Class Booking:

- Testing of this class is in application testing

Application Testing Plan

Test Suite 1:

Approach - All test cases with "**positive outputs**" and of every category are covered

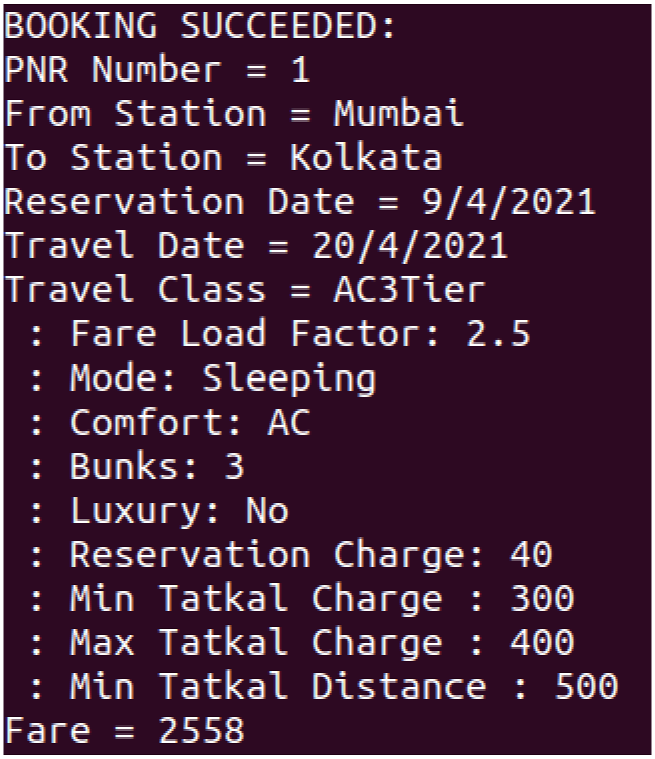
Test Cases -

- Test Case 1 -

Condition covered - Construction of booking of General category

Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then General class booking is created and it's output is considered for evaluation.

Golden Output



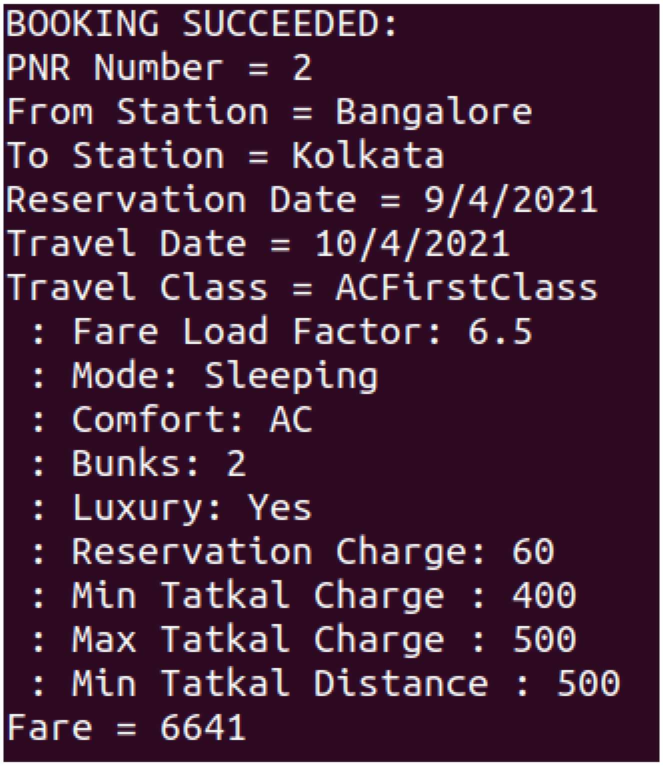
```
BOOKING SUCCEEDED:
PNR Number = 1
From Station = Mumbai
To Station = Kolkata
Reservation Date = 9/4/2021
Travel Date = 20/4/2021
Travel Class = AC3Tier
 : Fare Load Factor: 2.5
 : Mode: Sleeping
 : Comfort: AC
 : Bunks: 3
 : Luxury: No
 : Reservation Charge: 40
 : Min Tatkal Charge : 300
 : Max Tatkal Charge : 400
 : Min Tatkal Distance : 500
Fare = 2558
```

- Test Case 2 -

  Condition covered - Construction of booking of Tatkal category

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Tatkal class booking is created and it's output is considered for evaluation.

  Golden Output



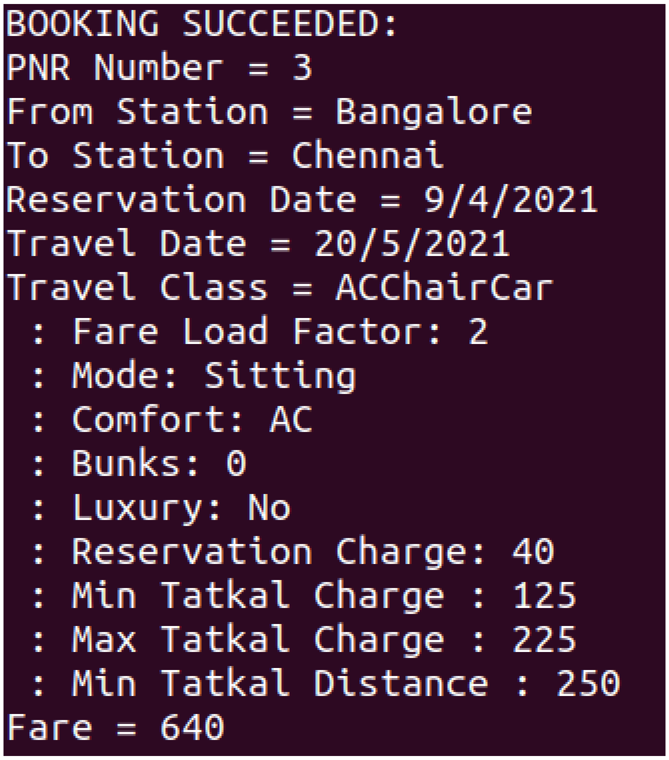
```
BOOKING SUCCEEDED:
PNR Number = 2
From Station = Bangalore
To Station = Kolkata
Reservation Date = 9/4/2021
Travel Date = 10/4/2021
Travel Class = ACFirstClass
 : Fare Load Factor: 6.5
 : Mode: Sleeping
 : Comfort: AC
 : Bunks: 2
 : Luxury: Yes
 : Reservation Charge: 60
 : Min Tatkal Charge : 400
 : Max Tatkal Charge : 500
 : Min Tatkal Distance : 500
Fare = 6641
```

- Test Case 3 -

  Condition covered - Construction of booking of Premium Tatkal category

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Premium Tatkal class booking is created and it's output is considered for evaluation.

  Golden Output



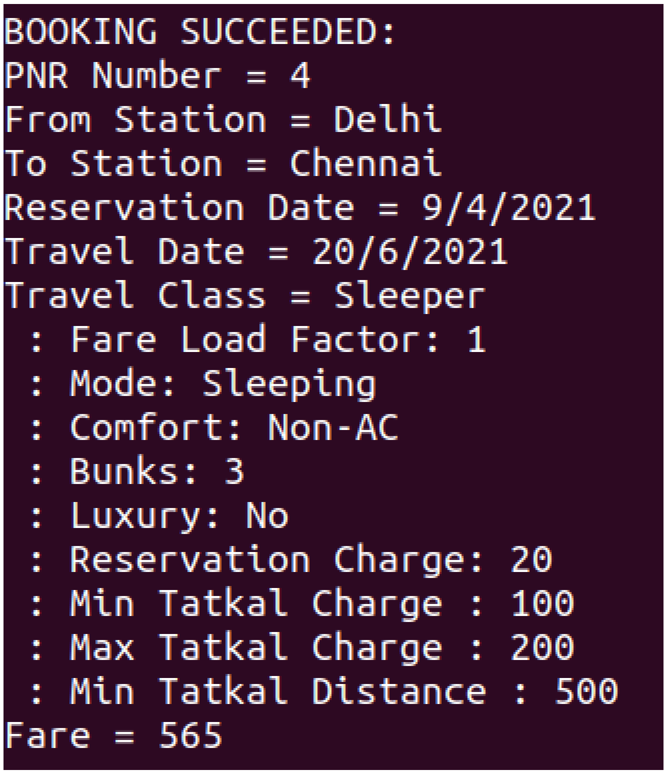
```
BOOKING SUCCEEDED:
PNR Number = 3
From Station = Bangalore
To Station = Chennai
Reservation Date = 9/4/2021
Travel Date = 20/5/2021
Travel Class = ACChairCar
 : Fare Load Factor: 2
 : Mode: Sitting
 : Comfort: AC
 : Bunks: 0
 : Luxury: No
 : Reservation Charge: 40
 : Min Tatkal Charge : 125
 : Max Tatkal Charge : 225
 : Min Tatkal Distance : 250
Fare = 640
```

- Test Case 4 -

  Condition covered - Construction of booking of Senior Citizen category

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Senior Citizen class booking is created and it's output is considered for evaluation.

  Golden Output



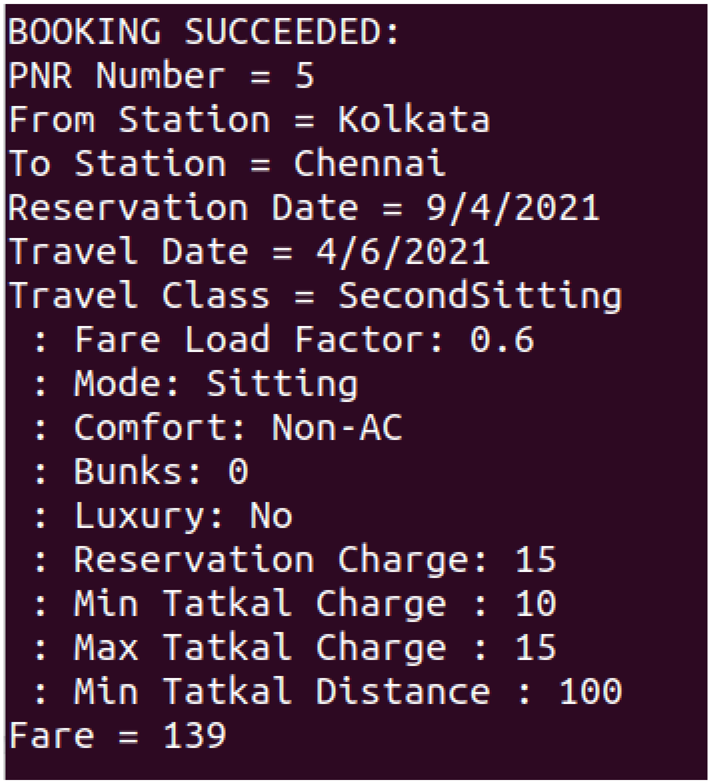
```
BOOKING SUCCEEDED:
PNR Number = 4
From Station = Delhi
To Station = Chennai
Reservation Date = 9/4/2021
Travel Date = 20/6/2021
Travel Class = Sleeper
 : Fare Load Factor: 1
 : Mode: Sleeping
 : Comfort: Non-AC
 : Bunks: 3
 : Luxury: No
 : Reservation Charge: 20
 : Min Tatkal Charge : 100
 : Max Tatkal Charge : 200
 : Min Tatkal Distance : 500
Fare = 565
```

- Test Case 5 -

  Condition covered - Construction of booking of Blind category

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Blind class booking is created and it's output is considered for evaluation.

  Golden Output



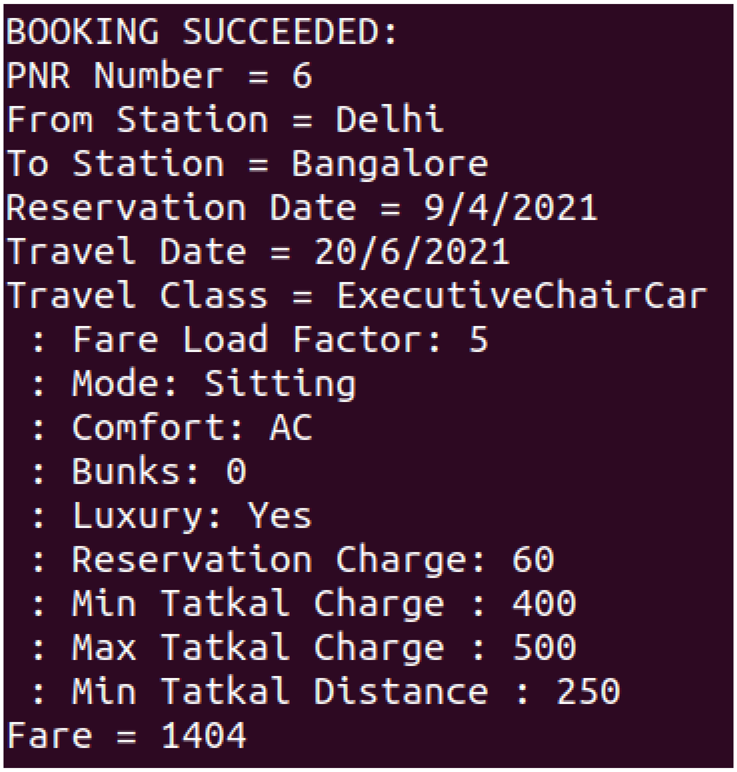
```
BOOKING SUCCEEDED:
PNR Number = 5
From Station = Kolkata
To Station = Chennai
Reservation Date = 9/4/2021
Travel Date = 4/6/2021
Travel Class = SecondSitting
 : Fare Load Factor: 0.6
 : Mode: Sitting
 : Comfort: Non-AC
 : Bunks: 0
 : Luxury: No
 : Reservation Charge: 15
 : Min Tatkal Charge : 10
 : Max Tatkal Charge : 15
 : Min Tatkal Distance : 100
Fare = 139
```

- Test Case 6 -

  Condition covered - Construction of booking of Orthopaedically Handicapped category

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Orthopaedically Handicapped class booking is created and it's output is considered for evaluation.

  Golden Output



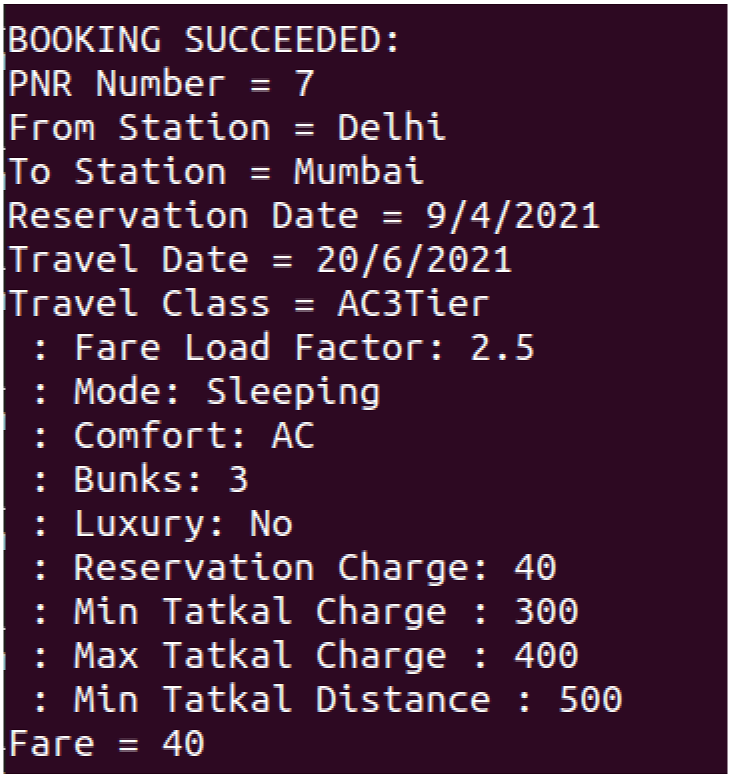
```
BOOKING SUCCEEDED:
PNR Number = 6
From Station = Delhi
To Station = Bangalore
Reservation Date = 9/4/2021
Travel Date = 20/6/2021
Travel Class = ExecutiveChairCar
 : Fare Load Factor: 5
 : Mode: Sitting
 : Comfort: AC
 : Bunks: 0
 : Luxury: Yes
 : Reservation Charge: 60
 : Min Tatkal Charge : 400
 : Max Tatkal Charge : 500
 : Min Tatkal Distance : 250
Fare = 1404
```

- Test Case 7 -

  Condition covered - Construction of booking ofCancer category

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then Cancer class booking is created and it's output is considered for evaluation.

  Golden Output



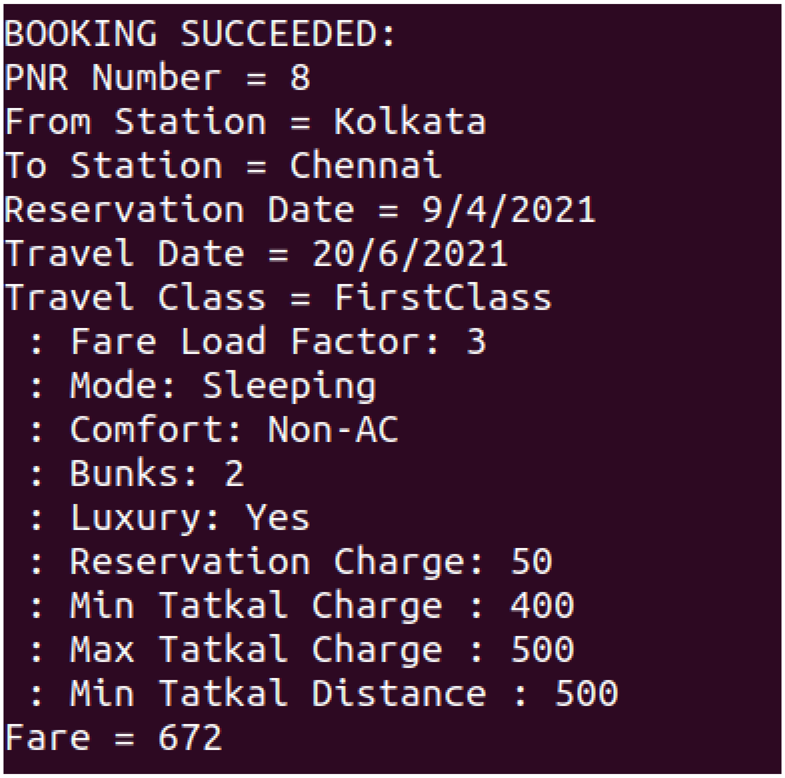
```
BOOKING SUCCEEDED:
PNR Number = 7
From Station = Delhi
To Station = Mumbai
Reservation Date = 9/4/2021
Travel Date = 20/6/2021
Travel Class = AC3Tier
 : Fare Load Factor: 2.5
 : Mode: Sleeping
 : Comfort: AC
 : Bunks: 3
 : Luxury: No
 : Reservation Charge: 40
 : Min Tatkal Charge : 300
 : Max Tatkal Charge : 400
 : Min Tatkal Distance : 500
Fare = 40
```

- Test Case 8 -

  Condition covered - Construction of booking of TB category

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then TB class booking is created and it's output is considered for evaluation.

  Golden Output



```
BOOKING SUCCEEDED:
PNR Number = 8
From Station = Kolkata
To Station = Chennai
Reservation Date = 9/4/2021
Travel Date = 20/6/2021
Travel Class = FirstClass
 : Fare Load Factor: 3
 : Mode: Sleeping
 : Comfort: Non-AC
 : Bunks: 2
 : Luxury: Yes
 : Reservation Charge: 50
 : Min Tatkal Charge : 400
 : Max Tatkal Charge : 500
 : Min Tatkal Distance : 500
Fare = 672
```

Test Suite 2 :-

Approach - All test cases with “**negative outputs**” and of every exceptions are covered

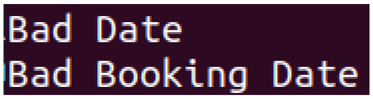
Test Cases -

- Test Case 1 -

  Condition covered - Construction of booking with invalid date

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong booking date and it’s output is considered for evaluation.

Golden Output



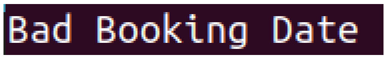
```
Bad Date
Bad Booking Date
```

- Test Case 2 -

  Condition covered - Construction of booking with past date

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong booking date and it's output is considered for

  Golden Output



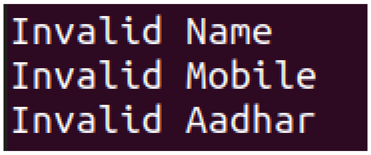
```
Bad Booking Date
```

- Test Case 3 -

  Condition covered - Construction of booking with invalid aadhar, mobile number and name

  Step Involved - Construct a passenger object using its constructor with wrong inputs. Then instance of class booking is created and it's output is considered for

  Golden Output



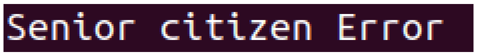
```
Invalid Name
Invalid Mobile
Invalid Aadhar
```

- Test Case 4 -

  Condition covered - Construction of booking with invalid age for senior citizen

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong age and it's output is considered for

  Golden Output



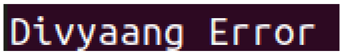
```
Senior citizen Error
```

- Test Case 5 -

  Condition covered - Construction of booking with wrong disabilty

  Step Involved - Construct a passenger object using its constructor with wrong disability. Then instance of class booking is created and it's output is considered for

  Golden Output

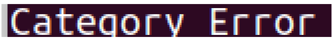
  ```
  Divyaang Error
  ```

- Test Case 6 -

  Condition covered - Construction of booking with wrong category

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong category and it's output is considered for

  Golden Output

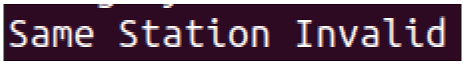
  ```
  Category Error
  ```

- Test Case 7 -

  Condition covered - Construction of booking with wrong station

  Step Involved - Construct a passenger object using its constructor with all valid inputs provided required by the constructor as inputs. Then instance of class booking is created and with wrong station and it's output is considered for

  Golden Output

  ```
  Same Station Invalid
  ```

The End